Zbornik 29. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2026

Zvezek C

Proceedings of the 29th International Multiconference

INFORMATION SOCIETY – IS 2026

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

DRAFT – NOT FOR PUBLICATION

https://is.ijs.si

7. oktober 2026 / 7 October 2026
Ljubljana, Slovenia

Urednika:

Dunja Mladenić
Department for Artificial Intelligence, Jožef Stefan Institute, Ljubljana

Marko Grobelnik
Department for Artificial Intelligence, Jožef Stefan Institute, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
https://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2026

Informacijska družba
ISSN 2630-371X

DRAFT – NOT FOR PUBLICATION

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2026

29. mednarodna multikonferenca *Informacijska družba* se odvija v času izjemne rasti umetne inteligence, njenih aplikacij in vplivov na človeštvo. Vsako leto bolj vstopamo v novo dobo, v kateri generativna umetna inteligenca ter drugi inovativni pristopi oblikujejo poti k superinteligenci in singularnosti, ki bosta krojili prihodnost človeške civilizacije. Naša konferenca je tako hkrati tradicionalna znanstvena in akademsko odprta, pa tudi inkubator novih, pogumnih idej in pogledov.

Letošnja konferenca poleg umetne inteligence vključuje tudi razprave o perečih temah današnjega časa: ohranjanje okolja, demografski izzivi, zdravstvo in preobrazba družbenih struktur. Razvoj UI ponuja rešitve za številne sodobne izzive, kar poudarja pomen sodelovanja med raziskovalci, strokovnjaki in odločevalci pri oblikovanju trajnostnih strategij. Zavedamo se, da živimo v obdobju velikih sprememb, ko je ključno, da z inovativnimi pristopi in poglobljenim znanjem ustvarimo informacijsko družbo, ki bo varna, vključujoča in trajnostna.

V okviru multikonference smo letos združili dvanajst vsebinsko raznolikih srečanj, ki odražajo širino in globino informacijskih ved: od umetne inteligence v zdravstvu, demografskih in družinskih analiz, digitalne preobrazbe zdravstvene nege ter digitalne vključenosti v informacijski družbi, do raziskav na področju kognitivne znanosti, zdrave dolgoživosti ter vzgoje in izobraževanja v informacijski družbi. Pridružujejo se konference o legendah računalništva in informatike, prenosu tehnologij, mitih in resnicah o varovanju okolja, odkrivanju znanja in podatkovnih skladiščih ter seveda Slovenska konferenca o umetni inteligenci.

Poleg referatov bodo okrogle mize in delavnice omogočile poglobljeno izmenjavo mnenj, ki bo pomembno prispevala k oblikovanju prihodnje informacijske družbe. »Legende računalništva in informatike« predstavljajo domači »Hall of Fame« za izjemne posameznike s tega področja. Še naprej bomo spodbujali raziskovanje in razvoj, odličnost in sodelovanje; razširjeni referati bodo objavljeni v reviji *Informatica,* s podporo dolgoletne tradicije in v sodelovanju z akademskimi institucijami ter strokovnimi združenji, kot so ACM Slovenija, SLAIS, Slovensko društvo Informatika in Inženirska akademija Slovenije.

Vsako leto izberemo najbolj izstopajoče dosežke. Letos je nagrado *Michie-Turing* za izjemen življenjski prispevek k razvoju in promociji informacijske družbe prejel **Denis Trček**, priznanje za raziskovalni dosežek leta pa **Jure Leskovec**. »Informacijsko limono« za najmanj primerno informacijsko storitev je prejel informacijski sistem za dolgotrajno oskrbo. »Informacijsko jagodo« za najboljši sistem ali storitev v letih 2025/2026 pa je prejela elektronska osebna izkaznica. Iskrene čestitke vsem nagrajencem!

Naša vizija ostaja jasna: prepoznati, izkoristiti in oblikovati priložnosti, ki jih prinaša digitalna preobrazba, ter ustvariti informacijsko družbo, ki koristi vsem njenim članom. Vsem sodelujočim se zahvaljujemo za njihov prispevek — veseli nas, da bomo skupaj oblikovali dosežke, ki bodo krojili usodo Slovenije in sveta.

Mojca Ciglarič, predsednica programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD TO THE MULTICONFERENCE INFORMATION SOCIETY 2026

The 29th International Multiconference *Information Society* takes place at a time of remarkable growth in artificial intelligence, its applications, and its impact on humanity. Each year we move further into a new era in which generative artificial intelligence and other innovative approaches are opening paths toward superintelligence and the singularity — two developments that will shape the future of human civilization. Our conference is at once a traditional scientific meeting, an academically open forum, and an incubator of new and bold ideas.

In addition to artificial intelligence, this year's conference also addresses pressing issues of our time: environmental protection, demographic challenges, healthcare, and the transformation of social structures. The development of AI offers solutions to many contemporary challenges and underscores the need for collaboration among researchers, professionals, and policymakers in shaping sustainable strategies. We are aware that we live in a period of profound change, when innovative approaches and deep knowledge are essential if we are to build an information society that is safe, inclusive, and sustainable.

This year's multiconference brings together twelve thematically diverse meetings that reflect the breadth and depth of the information sciences. They range from artificial intelligence in healthcare, demographic and family studies, the digital transformation of nursing, and digital inclusion in the information society, to research in cognitive science, healthy longevity, and education in the information society. They are joined by conferences on the Legends of Computing and Informatics, technology transfer, myths and truths of environmental protection, knowledge discovery and data warehouses, and, of course, the Slovenian Conference on Artificial Intelligence.

Alongside the papers, round tables and workshops will enable an in-depth exchange of views that will contribute substantially to shaping the future information society. *Legends of Computing and Informatics* serves as a national Hall of Fame for outstanding individuals in the field. We will continue to foster research and development, excellence, and collaboration. Extended papers will be published in the journal *Informatica*, building on a long-standing tradition and on cooperation with academic institutions and professional associations, including ACM Slovenia, SLAIS, the Slovenian Society Informatika, and the Slovenian Academy of Engineering.

Each year we single out the most distinctive achievements. This year the *Michie–Turing Award* for an outstanding lifetime contribution to the development and promotion of the information society was presented to **Denis Trček**, and the Award for Research Achievement of the Year to **Jure Leskovec**. The "Information Lemon" for the least appropriate information service went to the long-term care information system. The "Information Strawberry" for the best system or service in 2025/2026 went to the electronic identity card. Our warmest congratulations to all the award winners.

Our vision remains clear: to recognize, use, and shape the opportunities of digital transformation, and to create an information society that benefits all of its members. We thank all participants for their contributions. We look forward to shaping, together, achievements that will determine the fate of Slovenia and of the world.

Mojca Ciglarič, Chair of the Program Committee
Matjaž Gams, Chair of the Organizing Committee

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia
Sergio Campos-Cordobes, Spain
Shabnam Farahmand, Finland
Sergio Crovella, Italy

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Blaž Mahnič

Programme Committee

Mojca Ciglarič, chair
Bojan Orel,
Franc Solina,
Viljan Mahnič,
Cene Bavec,
Tomaž Kalin,
Jozsef Györkös,
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Mitja Luštrek
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Gašper Slapničar
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah
Niko Zimic
Rok Piltaver
Toma Strle
Tine Kolenik
Franci Pivec
Uroš Rajkovič
Borut Batagelj
Tomaž Ogrin
Bojan Blažica
Matjaž Kljun
Robert Blatnik
Erik Dovgan
Špela Stres
Anton Gradišek
Borut Trpin
Anka Slana Ozimič

KAZALO / TABLE OF CONTENTS

Odkrivanje znanja in podatkovna skladišča - SiKDD / Data Mining and Data Warehouses - SiKDD 1

PREDGOVOR / FOREWORD 3

PROGRAMSKI ODBORI / PROGRAMME COMMITTEES 5

Exploring Feedback-Aware Retrieval for IT Support First-Reply Generation / Jeršek Domen, Kažič Blaž, Kenda Klemen 7

The Changing Slovenian Smellscape, 1870–1919: Measuring Olfactory Discourse / Novalija Inna, Brank Janez, Mladenić Dunja, Grobelnik Marko 11

Explaining Headline A/B Test Outcomes with Sparse Autoencoder Features / Pesjak Drejc, PI2, PI3 15

Same Labels, Smaller Bill: An Ablation Study of LLM Claim Classification in Cross-Source News / Trajkov Georgi, Škrlj Blaž, Grobelnik Marko, Mladenić Dunja 19

A Hybrid Lexicon-LLM Approach to Skill Extraction from Job Advertisements / Mečkaroska Natalija, Trajkov Georgi 23

Benchmarking Machine-Learning Forecasts of Low-Magnitude Seismicity in the Italy–Slovenia Border Region / Topal Oleksandra, Calcina Erik, Telesca Luciano, Mladenić Dunja 27

The Effect of Label Only vs. Label Plus Description: Comparing GLiNER1 and GLiNER2 for Clinical Entity Recognition Across Five Languages / Sittar Abdul, Novak Erik, Calcina Erik 31

Data Efficiency of GLiNER: A Multilingual and Multi-Scale Learning Curve Study / Kokalj Nina, Calcina Erik, Sittar Abdul, Novak Erik 35

Comparing Pseudo-Labeling Criteria for Active Learning on a Service-Desk Dataset / Klančič Rok, Kenda Klemen 39

LLM-Generated Song Lessons: Perceived Usefulness and Shared-Cache Cost Analysis / Trajkov Georgi 43

Reliable but Not Right: Automated Evaluation of Generated R&D Ideas / Mladenić Grobelnik Adrian 47

Graphs of n-grams for Text Generation and Characteristic Phrase Extraction in the Domain of Smell / Komarova Nadezhda, Novalija Inna 51

QFS-Composer: Query-Focused Summarization Pipeline for Less Resourced Languages / Đuranović Vuk, Robnik-Šikonja Marko 55

Beyond Named Citations: A Corpus of Resolved Cross-References in EU Legislation / Vokač Jakob, Grobelnik Marko, Mladenić Dunja 61

Towards Scientific Contradiction Mining: Topic Discovery and Claim Extraction from AI Literature / Natev Andrej, Motamedi Elham, Novalija Inna 65

Chronos-2 for Spacecraft Telemetry Forecasting / Klun Žiga, Kocev Dragi, Brence Jure 69

Biomedical Relation Extraction: What Lies Between the Entities? / Shpanko Nadiia, Calcina Erik, Novak Erik 73

The Role of Artificial Intelligence in Early Moisture Risk Detection for Wooden Buildings / Piiarska Kristina, Tošić Aleksandar, Brunet Pau 77

E3 Atlas: An Autonomous LLM Agent for Curating and Serving a Departmental Knowledge Base / Pesjak Drejc, Trajkov Georgi, Rei Luis, Mladenić Dunja 81

Evaluating Large Language Model-Generated Explanations for Defective-Part Prediction in Injection Moulding Using Feature-Attribution Evidence / Hosseini Seyed Iman, Seghesio Lorenzo, Gryllias Konstantinos, Mladenić Dunja 85

Learning Molecular Representations from SMILES for Acute Toxicity Prediction / Cholakova Simona, Vake Domen, Tošić Aleksandar 89

Logical Anomaly Detection Using Segmentation Transformers / Liang Yunfei, Fučka Matic, PI3 93

Posterior Contraction Rate and Roadmap to Semiparametric Bernstein-von Mises Theorem for Bayesian Causal Forests / Srakar Andrej 97

Validation of Proper Noun Lexemes in the Digital Dictionary Database of Slovene Using Wikidata and Large Language Models / Čibej Jaka 101

Comparative Analysis of Feature Selection and Imbalance Handling for Semiconductor Defect Detection on the SECOM Dataset / Uku Sidorela, Kenda Klemen 105

Operationalising Active Learning for Power Grid Security Assessment: A Human-in-the-Loop Retraining Service for a Digital Twin Platform / Leskovec Gašper, Mylonas Costas, Kenda Klemen 109

Od kvantnega strojnega učenja do kvantne umetne inteligence: sistematični meta-pregled / Brodschneider Aleksandar, Gams Matjaž 113
From Explainability to Accountability: XAI and Fundamental Rights Impact Assessments under the EU AI Act / Bobnar Lovro 117
Influencing Sentiment in LLM Output via Token Selection Manipulation / Praprotnik Kaja, Hrovatin Niki, Vičič Jernej, Vake Domen 122
From Label Agreement to Downstream Performance: Evaluating LLM-Generated Training Data / Mammana Dario, Kralj Novak Petra 126
Short Term Time-Series Forecasting Framework / Kreft Gašper, Osojnik Aljaž, Žnidaršič Martin, Ženko Bernard 130
Short-Term Load Forecasting at the Device Level: Are Specialized Deep-Learning Approaches Worth the Effort? / Zmazek Gal, Osojnik Aljaž, Žnidaršič Martin, Ženko Bernard 134
Data-Driven and Model-Based Causal Discovery in Sparse Multivariate Time Series: A Comparative Framework for Sustainable Development Systems / Pita Costa Joao, Srakar Andrej, Topal Oleksandra, Šturm Jan 138
Legal Categories from Incident Reports: Evaluating 14 LLMs under the EU AI Act/ Kofilovski Stefan, Trajkov Georgi 142
Balancing Sustainable Development Goals through Orchestrating Autonomous Multi-Agent Architecture / Guruvayur Sivaramakrishnan R, Achuthan Krishnasree, Pita Costa Joao 146
Accurate but Uninformative: Checking an LLM Assessment Pipeline Without Ground Truth / Macak Martin, Grobelnik Marko 150
Combining Supervised and Unsupervised Learning on SMART Data for Hard Drive Failure Assessment / Petek Luka, Podgorelec Vili 154
Advertising Is Not Noise: Sponsored Speech Outranks New Terminology in Automatic Term Extraction / Macak Martin 158

Indeks avtorjev / Author index 163

Zbornik 29. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2026

Zvezek C

Proceedings of the 29th International Multiconference

INFORMATION SOCIETY – IS 2026

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

DRAFT – NOT FOR PUBLICATION

https://is.ijs.si

7. oktober 2026 / 7 October 2026
Ljubljana, Slovenia

PREDGOVOR

Tehnologije, ki se ukvarjajo s podatki so močno napredovale. Iz prve faze, kjer je šlo predvsem za shranjevanje podatkov in kako do njih učinkovito dostopati, se je razvila industrija za izdelavo orodij za delo s podatkovnimi bazami in velikimi količinami podatkov, prišlo je do standardizacije procesov, povpraševalnih jezikov. Ko shranjevanje podatkov ni bil več poseben problem, se je pojavila potreba po bolj urejenih podatkovnih bazah, ki bi služile ne le transakcijskem procesiranju ampak tudi analitskim vpogledom v podatke. Pri avtomatski analizi podatkov sistem sam pove, kaj bi utegnilo biti zanimivo za uporabnika – to prinašajo tehnike odkrivanja znanja v podatkih (knowledge discovery and data mining), ki iz obstoječih podatkov skušajo pridobiti novo znanje in tako uporabniku nudijo novo razumevanje dogajanj zajetih v podatkih. Slovenska KDD konferenca SiKDD, pokriva vsebine, ki se ukvarjajo z analizo podatkov in odkrivanjem znanja v podatkih: pristope, orodja, probleme in rešitve.

Letošnja konferenca SiKDD 2026 združuje 38 prispevkov, ki obravnavajo velike jezikovne modele in generativno umetno inteligenco, obdelavo naravnega jezika in semantične tehnologije, napovedovanje časovnih vrst in analitiko senzorskih podatkov, razložljivo in zaupanja vredno umetno inteligenco ter uporabo teh pristopov na področjih, kot so energetika, proizvodnja, biomedicina, pravo in trajnostni razvoj.

FOREWORD

Data driven technologies have significantly progressed. The first phases were mainly focused on storing and efficiently accessing the data, resulted in the development of industry tools for managing large databases, related standards, supporting querying languages, etc. After the initial period, when the data storage was not a primary problem anymore, the development progressed towards analytical functionalities on how to extract added value from the data; i.e., databases started supporting not only transactions but also analytical processing of the data. In automatic data analysis, the system itself tells what might be interesting for the user - this is brought about by knowledge discovery and data mining techniques, which try to obtain new knowledge from existing data and thus provide the user with a new understanding of the events covered in the data. The Slovenian KDD conference SiKDD covers topics dealing with data analysis and discovering knowledge in data: approaches, tools, problems and solutions.

This year, SiKDD 2026 brings together 38 papers addressing large language models and generative AI, natural language processing and semantic technologies, time-series forecasting and sensor data analytics, explainable and trustworthy AI, and applications of these approaches in domains such as energy, manufacturing, biomedicine, law and sustainable development.

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Janez Brank, Jožef Stefan Institute, Ljubljana

Erik Calcina, Jožef Stefan Institute, Ljubljana

Jasminka Dobša, Faculty of Organization and Informatics, University of Zagreb

Alenka Guček, Jožef Stefan Institute, Ljubljana

Branko Kavšek, University of Primorska, Koper

Klemen Kenda, Qlector, Ljubljana

Bojana Mikelenić, Faculty of Humanities and Social Sciences, University of Zagreb

Elham Motamedi Mohammadabadi, Jožef Stefan Institute, Ljubljana

Irena Nančovska Šerbec, Faculty of Education, University of Ljubljana

Erik Novak, Jožef Stefan Institute, Ljubljana

Inna Novalija, Jožef Stefan Institute, Ljubljana

Joao Pita Costa, Quintelligence, Ljubljana

Jože Rožanec, Jožef Stefan Institute, Ljubljana

Abdul Sittar, Jožef Stefan Institute, Ljubljana

Luka Stopar, SolvesAll, Ljubljana

Blaž Škrlj, Teads, Ljubljana

Jan Šturm, Jožef Stefan Institute, Ljubljana

Oleksandra Topal, Jožef Stefan Institute, Ljubljana

Exploring Feedback-Aware Retrieval for IT Support First-Reply Generation

Domen Jeršek
domenjersek@gmail.com
Jožef Stefan Institute
Slovenia

Blaž Kažič
blaz.kazic@ijs.si
Jožef Stefan Institute
Slovenia

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Slovenia

Abstract

We examine *when* adding feedback signals to retrieval-augmented generation (RAG) helps the generated first reply to IT support tickets and when it hurts. Using a FAISS-based pipeline on 250 tickets under strict leave-one-out isolation, we compute a Laplace-corrected Bayesian popularity lift from 102,722 LLM-judge-scored (query, candidate) pairs drawn from an approximately 1,600-ticket knowledge base. The lift replaces the top-5 FAISS ranking (98 % of tickets change candidates, median promoted rank 17). In an exploratory analysis, we find that feedback popularity and semantic similarity are orthogonal ($r = 0.06$), and an oracle signal exists: feedback helps a weak baseline (+0.21 in the weakest decile) and hurts a strong one (-0.09; $r = -0.41$). We test cosine-based gating (it fails because the retrieval-to-answer path, $r = 0.24$, is too weak) and pool-level filtering (M1 global, M2 team, M3 type, M4 team∩type). Pool analysis reveals why some pools benefit: query diversity ($r = -0.38$) and procedural mismatch ($r = 0.21$) are candidate gating features observable before generation. A pool-based oracle gate reaches +0.017 in-sample, motivating a learned gating model as the natural next step.

Keywords

IT support, retrieval-augmented generation, LLM-judge feedback, retrieval replacement, exploratory analysis

1 Introduction

Enterprise IT-support teams answer thousands of tickets, many nearly identical to tickets answered before. RAG is appealing: retrieve the most similar historical tickets and let a language model write the first reply. The implicit assumption is that *textual* similarity tracks *procedural* similarity.

That assumption can fail. A ticket saying *"I cannot open* `.ica` *files in Citrix"* retrieves Citrix *troubleshooting* neighbours that share its vocabulary. But the correct reply is not a troubleshooting guide; it is a form redirect to a software-installation request. The right template lives among less similar neighbours (we return to this exact ticket in §5).

Feedback offers a way out: if some past replies are known to be more useful, a candidate whose reply was rated useful could be promoted even if textually dissimilar. Our prior work [1] demonstrated a RAG pipeline for this domain; here we add LLM-judge feedback as a retrieval modifier and ask a diagnostic question:

> *When does adding feedback to retrieval improve the generated first reply, and when does it hurt — and can we tell the difference before we generate anything?*

This is an exploratory paper structured around three research questions. **RQ1**: Does feedback-aware retrieval improve first-reply generation over semantic retrieval alone? We compare four feedback-pooling scopes (M1 global, M2 team, M3 type, M4 team∩type). **RQ2**: Under what conditions does feedback help or hurt? We test baseline quality, team/type membership, procedural mismatch, and within-pool query diversity as explanatory factors. **RQ3**: Can the benefit be predicted before generation? We sweep retrieval-margin thresholds, pool-identity gates, and text-derived features, finding that simple retrieval-only signals are insufficient. The central finding remains: feedback and semantic similarity are orthogonal signals ($r = 0.06$), making feedback-driven retrieval an unconditional trade-off.

2 Related Work

RAG [3] combines parametric LLM knowledge with external retrieval. Dense retrieval with sentence embeddings [5, 6] and FAISS indexing [2] provide efficient semantic search. Feedback-for-retrieval is under-explored compared to feedback-for-generation (RLHF [4]). We use structured procedural ratings from an LLM judge [7] and ask the *conditional* question of when feedback helps. Jeršek et al. [1] built the same RAG pipeline for this domain without feedback; we extend it with LLM-judge feedback signals. The finding that retrieval confidence weakly predicts answer quality aligns with the broader observation that retrieval-time similarity metrics routinely under-predict downstream accuracy.

3 Data

Data flow. After cleaning 1,989 raw tickets for missing data, the KB contains **1,596 tickets** (title, description, historical first reply). A FAISS index is built over this full set. **250 query tickets** are randomly selected for evaluation; a query's own ticket is excluded from its candidate pool during evaluation (standard LOO).

KB pool distribution. The KB spans 37 resolver teams and 5 first-reply intent labels (derived by co-occurrence clustering of resolver-team×Service-field-value pairs). `other` (747 tickets, 47 %) and `vpn_request` (586, 37 %) dominate; `software_request` (133, 8 %), `admin_rights` (117, 7 %), and `email_support` (13, <1 %) make up the remainder. Across 75 team×intent combinations, only 15 % are singletons — broader than the query set (38 % singletons), which later explains why M4 starves for evidence (§6.2).

Feedback database. An LLM judge (gpt-4o-mini, temperature 0) scored each (query, candidate) pair. It received the query title, description, the *ground-truth expected first reply*, and candidate tickets with their first replies, returning a score in $[0, 1]$. This LLM-judge feedback simulates expert evaluation: the judge acts as a domain expert, comparing candidate replies against the ground-truth reply written by the original resolver. **Importantly, the judge evaluated retrieval candidates (not generated answers)**: it compared each candidate ticket's historical first reply to the expected reply, producing a retrieval-level quality

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.96

score. Scores from approximately 225 batch evaluation runs were merged into 102,722 scored rows (250 queries, approximately 1,580 physical candidates, 58,318 unique query–candidate pairs). A candidate is positive (mean ≥ 0.80) or negative (mean ≤ 0.40); the middle band is ignored. Repeated scoring does not inflate n: the per-pair mean determines the label. Because both the judge and the cosine evaluation metric reference this ground truth, we validate all findings under ROUGE-L and BERTScore (§5), confirming they are not an artefact of scoring against the evaluation target.

4 Methodology

4.1 Pipeline

The system is a two-stage pipeline (Figure 1). **Retrieval:** Title and description are embedded with all-MiniLM-L6-v2 [6, 5]. The top-5 most similar tickets are retrieved from a FAISS index [2]. The *baseline* path returns these directly; the *feedback* path **replaces** the ranking using FAISS score plus a feedback lift. **Generation:** The top-5 are formatted as few-shot examples (title + first reply) and passed to gpt-5.6-luna via OpenRouter.

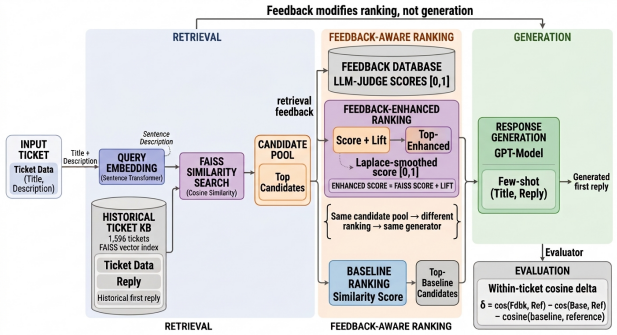


Figure 1: System architecture. Baseline and feedback differ only in the ranking of the same candidate pool.

4.2 The lift formula

For candidate c and query q, let pos and neg be positive/negative counts, $n = pos + neg$. The lift is:

$$p = \frac{pos + \alpha}{pos + neg + \alpha + \beta}, \quad \text{lift} = (p - 0.5) \cdot \min(1,\, n/2) \cdot m,$$

clamped to $\pm C, with\ \alpha = \beta = 1.0$, $m = 0.80$, $C = 0.20$. The enhanced score is $\text{FAISS}(c, q) + \text{lift}(c)$; top-5 by enhanced score become the feedback retrieval.

We define **fidelity** as the Pearson correlation between the continuous mean judge score and the assigned lift, measuring how faithfully the formula preserves score ordering. We compared three candidate formulas on 24,111 lift values from 100 sampled queries (sufficient for formula selection, since the per-candidate score distribution stabilises well below 250 queries): Laplace, tanh shrinkage, and Bayesian lower confidence bound (LCB). Laplace has the widest dynamic range ($\sigma = 0.090$ vs. tanh $\sigma = 0.024$) and the highest fidelity to the underlying scores ($r = 0.85$ vs. tanh 0.72, LCB 0.68). It is weakly driven by observation count ($r = 0.19$ to n vs. tanh 0.31, LCB 0.45). The alternatives fail structurally: tanh compresses nearly all values toward zero, making candidates effectively indistinguishable; Bayesian LCB collapses 82 % of lifts to exactly zero, destroying ranking resolution. A limitation: quantization into discrete values ($0, \pm 0.067, \pm 0.133, \pm 0.200$) due to integer counts and $\min(1, n/2)$. The parameters $m = 0.80$, $C = 0.20$, and thresholds 0.80/0.40 were fixed a priori.

Filtering scopes (M1–M4). The lift can be pooled at four granularities:

- **M1 (global):** all 250 queries contribute — primary scope, largest n.
- **M2 (team-only):** only queries from the same resolver team.
- **M3 (type-only):** only queries with the same intent label.
- **M4 (team∩type):** strict intersection — smallest pools, starves for 33 % of tickets where no feedback lift exists.

M2–M4 use known team/type labels to isolate scope effects without classifier noise.

4.3 Evaluation

Evaluation is strict leave-one-out (250 folds). For each query, its own feedback rows and ticket are removed from the pool (for M2–M4 this means the query's own vote is excluded from its team or type pool). The candidate pool and generator are identical between baseline and feedback; only the ranking differs. The primary metric is within-ticket cosine delta: $\delta(t) = (fb_t, ref_t) - (base_t, ref_t)$. Cosine correlates $r > 0.85$ with ROUGE-L and BERTScore.

5 Exploratory Analysis

To answer RQ2, we first characterise how feedback changes retrieval, then examine the conditions under which the difference is beneficial.

5.1 Feedback replaces retrieval

Feedback changes at least one candidate for **98 % of tickets** (mean overlap 1.71/5, Figure 2).

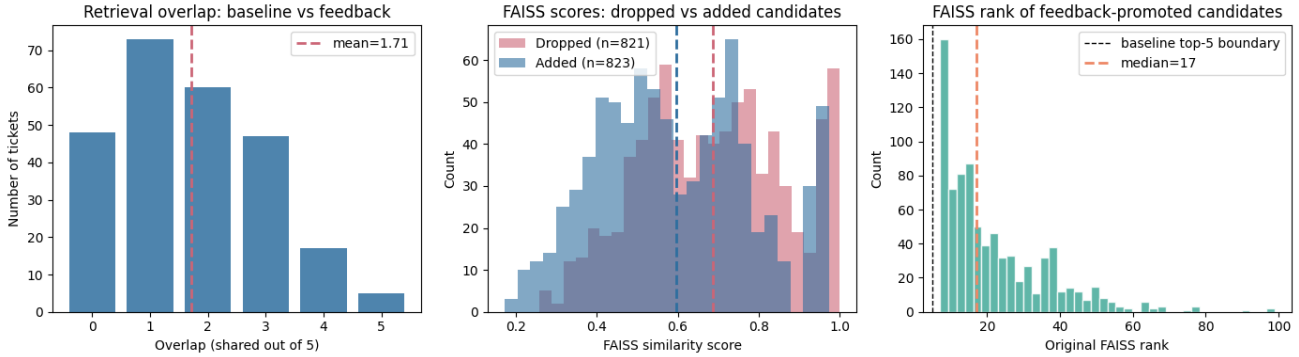


Figure 2: Feedback replaces retrieval, not reorders it. Promoted candidates have lower FAISS scores and come from deep in the FAISS ranking (median rank 17).

Promoted candidates have *lower* semantic similarity than those they replace (added 0.60 vs. dropped 0.69) and come from deep in the FAISS ranking (median rank 17). This happens because feedback popularity — $(pos - neg)/(pos + neg + 1)$ — and FAISS similarity are uncorrelated ($r = 0.058$ across 1,250 pairs). Only 5 % of candidates are in the top quarter of both.

5.2 Oracle signal and retrieval-vs-answer relationship

We define **oracle** as any post-generation quantity that distinguishes tickets where feedback helps from those where it harms. Figure 3 (left) shows the key finding: delta exhibits a clear negative trend against baseline answer cosine ($r = -0.41$). The weakest decile gains **+0.21** (76 % improved), the strongest loses **−0.09** (16 % improved). Feedback helps when the baseline is weak and hurts when it is strong. Before testing cosine-based gating, we examine the retrieval-to-answer-to-delta chain that any gate must traverse. Figure 3 (right) shows the first link: top-1 FAISS similarity predicts baseline answer cosine at $r = 0.24$ (6 % of variance). Combined with the oracle's answer-to-delta link ($r = -0.41$, 17 % of variance), the indirect retrieval-to-delta path is $r \approx 0.24 \times (-0.41) \approx -0.10$, while the *direct* retrieval-to-delta correlation measured on the data is even weaker ($r \approx -0.03$). In variance terms, a retrieval-only gate captures roughly 1 % ($r^2 = 0.01$) of the signal the oracle captures ($r^2 = 0.17$). This is a **structural gap** — not a tuning failure, but a fundamental information bottleneck: the quantity that predicts feedback benefit (baseline answer quality) is itself only weakly predictable from pre-generation retrieval signals.

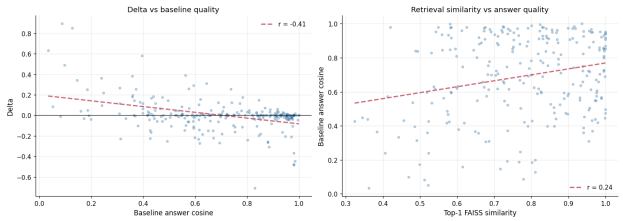


Figure 3: Left: delta vs. baseline answer quality — a clear negative trend ($r = -0.41$): feedback helps weak baselines and hurts strong ones. Right: retrieval similarity predicts answer quality only weakly ($r = 0.24$, 6 % variance), explaining why cosine-based gating fails.

Caveat. The baseline-cosine vs. delta relationship is partly mechanical (baseline cosine appears on both sides of delta). The same direction holds under ROUGE-L (D1 +0.18, D10 −0.09, $r = -0.31$) and BERTScore (D1 +0.02, D10 −0.02, $r = -0.31$).

5.3 Worked examples

These are the most extreme cases in the dataset (mean $\delta = +0.004$), chosen to illustrate the mechanism rather than to represent the typical ticket.

R-163 — the rescue ($\delta = +0.85$). This is the `.ica`-file ticket from the introduction. A user cannot open an `.ica` file in Citrix. The baseline retrieves troubleshooting neighbours and produces a generic reply (cosine 0.13). The correct form-redirect candidate was already at FAISS rank 4; feedback re-ranked within the same five, and the generator reproduced the form almost verbatim (cosine 0.98).

R-143 — the disruption ($\delta = -0.71$). A Salesforce travel-requisition request. The baseline produces a sound reply about network authorisation (cosine 0.83). Feedback promotes upvoted Salesforce candidates from deep ranks about different procedures — user permissions and onboarding. The generator switches to the wrong template (cosine 0.12). `Salesforce` is the most consistently feedback-hostile team (mean $\delta = -0.116$, §5): heterogeneous sub-procedures pooled under one label.

5.4 Pool delta distributions

Team-level mean deltas (Figure 4) span a wide range: `Group` benefits (+0.048) while `Salesforce` is harmed (−0.116). The team delta rank-order is cross-scope stable ($r > 0.55$ across M1–M4), meaning the pool-level pattern is not an artefact of the global scope. Type-level deltas show the same structure: `admin_rights` (+0.024) and `software_request` (+0.028) consist of reusable procedure templates where popularity tracks correctness; `email_support` (−0.074) is feedback-hostile.

These stable deltas suggest a causal hypothesis: same-team feedback excludes procedurally-unrelated votes, making it more relevant than the global pool.

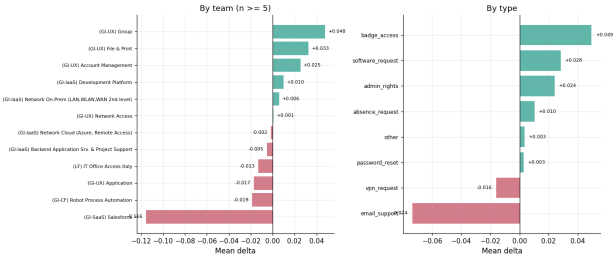


Figure 4: Mean feedback delta by team (left, teams with ≥ 5 tickets) and by ticket type (right). Teal: feedback benefits; red: feedback harms.

6 Modeling

6.1 Cosine-based gating

To answer RQ3 — whether the benefit can be predicted before generation — we test whether deployable retrieval-score proxies can approximate the oracle. An oracle gate — applying feedback only to tickets below a baseline-quality threshold — concentrates the benefit: the best threshold (0.6) routes feedback to 88/250 tickets for a mean delta of **+0.026**, versus +0.004 always-on.

However, the deployable proxies fail. A 5-fold cross-validated sweep over top-1 FAISS, top-5 FAISS, retrieval margin, and top5-minus-top1 thresholds finds no signal distinguishable from always-on feedback: the best signal (top5-minus-top1) reaches +0.006 versus +0.004 always-on, within cross-validation noise (±0.03). As shown in §5, the retrieval-to-delta signal captures only ∼1 % of the oracle's variance. Rescues ($\delta > +0.02$) and disruptions ($\delta < -0.02$) have near-identical top-1 FAISS scores (0.76 vs. 0.75). Simple similarity-threshold routing fails.

6.2 Pool filtering: M2, M3, M4

To answer RQ3 for pool-scoped evidence, we narrow the *feedback pool*. Team deltas being wide and stable suggests that feedback computed from team-only or type-only evidence might outperform global feedback. We test three additional scopes: M2 (team-only), M3 (type-only), and M4 (team∩type). M4 yields the largest

mean delta at +0.010 (vs. M1's +0.004), but starves for 33 % of tickets where no nonzero-lift candidates exist (§3). M2 and M3 are intermediate: +0.003 and −0.002 respectively.

6.3 Pool analysis: why some pools benefit

Returning to the pool delta distributions from §5, we test three post-hoc hypotheses. Structural score does not predict delta ($r = 0.06$). **Procedural mismatch** — a problem-worded query whose answer is a form redirect — is the strongest ticket-level signal: mismatch tickets gain +0.082 vs. −0.011 ($r = 0.21$). At the pool level this correlation reverses ($r \approx -0.59$, $n = 12$ teams): no single pattern explains both levels, motivating a learned gate. **Query diversity** — mean pairwise Jaccard similarity of ticket descriptions within a pool — predicts pool-level delta ($r = -0.38$, $n = 12$ teams): diverse pools (Group: VPN, passwords) are where FAISS retrieves the wrong procedure and feedback rescues; homogeneous pools (permission-change templates) are where FAISS is already right and feedback disrupts.

A naive in-sample team-identity gate reaches +0.017 (154/250 tickets). **Critical disclaimer:** these in-sample oracle numbers are circular; they motivate learning generalisable pool-level features, not a deployable rule.

7 Results

Table 1 consolidates all methods with bootstrap 95 % confidence intervals. All always-on methods (M1–M4 and the cosine gate) have CIs that cross zero; only the oracle gate (baseline answer quality < 0.6, 88/250 tickets) sits clearly above zero. The pool-based oracle gate (team-identity, in-sample) reaches +0.017 but is expressly circular.

Table 1: Method comparison (gpt-5.6-luna, $n = 250$, 10,000 bootstrap resamples). All CIs cross zero except the oracle gate. Pool-based oracle gate (§6.3) is in-sample only.

Method	Mean δ	95 % CI
Baseline (FAISS only)	0.000	—
M1 global	+0.004	[−0.016, +0.025]
Cosine gate*	+0.006	[−0.012, +0.026]
M2 team-only	+0.003	[−0.019, +0.026]
M3 type-only	−0.002	[−0.021, +0.018]
M4 team∩type	+0.010	[−0.006, +0.029]
Oracle gate†	+0.026	[+0.011, +0.043]
Pool-based oracle‡	+0.017	in-sample only

*Best 5-fold CV threshold (top5 minus top1 FAISS); within CV noise of always-on. †Baseline answer quality < 0.6 (88/250 tickets). ‡Team-identity: route feedback only to teams with positive mean delta, chosen on the same 250 tickets — **circular**, for directional intuition only.

8 Discussion

Limitations. Single enterprise, single evaluation run, 250 query tickets. LLM-judge feedback is a stronger signal than deployable human ratings: the judge sees the ground-truth reply and the evaluation metric measures similarity to it. Cosine similarity to a single reference reply is a narrow measure of usefulness; a ticket may admit more than one valid reply. Pool hypotheses are post-hoc and require held-out validation. Parameters were fixed a priori but sensitivity to m and C is left to future work. Team and type gating priors may not transfer to other organisations. Multiple-comparison corrections would not change the findings (the only CI above zero is the oracle gate).

Future work. Given the structural gap between retrieval signals and feedback benefit (§5), a learned gating model is the natural next step. Such a model could combine pool-level query diversity, procedural mismatch indicators, retrieval margin, and feedback-evidence strength to predict *which* filtering scope (M1–M4, or an adaptive pool) will perform best for each incoming ticket, using only pre-generation features. This would test whether the ∼0.017 pool-based oracle ceiling generalises beyond the in-sample analysis presented here.

9 Conclusion

Adding LLM-judge feedback to retrieval helps when the baseline retrieval is procedurally wrong and hurts when it is already right — the two signals are orthogonal. Simple retrieval-score proxies do not usefully gate this effect; the retrieval-to-answer-to-delta chain is too indirect. Narrowing the feedback pool by team or type (M2–M4) modestly improves the best case but starves for evidence. However, pool-level properties — query diversity ($r = -0.38$) and procedural mismatch ($r = 0.21$) — are observable before generation and predict benefit. The natural next step is a learned gating model that combines these pool-level features with evidence strength to decide *which* feedback pool to use *for which* ticket.

10 Acknowledgements

This work was supported by the European Union-funded project HumAIne [grant number 101120218]. The authors acknowledge the use of LLMs for language optimisation and feedback scoring. All conceptual frameworks, analyses, and interpretations remain the sole responsibility of the authors.

References

[1] Domen Jeršek, Klemen Kenda, Rok Klančič, and Matteo Frattini. 2025. Automated first-reply generation for it support tickets using retrieval-augmented generation and multi-modal response synthesis. In *Proceedings of the 28th International Multiconference Information Society (IS 2025)*. doi:10.70314/is.2025.sikdd.19.

[2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7, 3, 535–547.

[3] Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.

[4] Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.

[5] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, 3982–3992.

[6] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*. Vol. 33, 5776–5788.

[7] Lianmin Zheng et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*. Vol. 36.

The Changing Slovenian Smellscape, 1870–1919: Measuring Olfactory Discourse

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Janez Brank
janez.brank@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Counts of smell references in a historical corpus rise when the corpus itself grows, so raw frequencies cannot tell whether a culture talked about smell more often. We re-examine 366,333 passages of Slovene print from 1870–1919, previously processed with a curated smell lexicon and an LLM reader, and correct three biases without any manual annotation: exposure, word sense, and hallucination. Sense confirmation (the rate at which the LLM confirms a lexicon match in the same passage) separates olfactory uses of polysemous entries from non-olfactory ones and recovers explicit Slovene olfactory terms unaided (AUC 0.92, split-half $r = 0.98$). It shows that 52% of all keyword matches belong to term types the LLM corroborates in under 2% of their occurrences, dominated by polysemy such as *pot* ('sweat' / 'path'). After correction the apparent growth of olfactory discourse disappears: raw counts rise 5.5-fold while the corpus grows 6.5-fold, and the exposure-corrected rate is flat ($\tau = -0.07$, $p = 0.46$). What changes is composition. Food and drink rise from 19% to 32% of mentions and industry from 12% to 18%, while nature falls from 41% to 24%. Contrary to the deodorisation thesis, smell talk becomes *more* pleasant, and the drift is carried by a doubling in the share of purchasable scented goods — a commercialisation of the printed smellscape rather than a retreat from odour.

Keywords

Digital Humanities, Cultural Heritage, LLM, Text Mining, Sensory History

1 Introduction

Olfactory history has an evidence problem. Smell leaves no artefact, so its past is reconstructed almost entirely from what people wrote about it [4], and large digitised newspaper and book collections are now the obvious place to look. But a digitised collection is not a sample of a culture; it is a sample of what was printed, survived, was scanned, and passed OCR. All four grow steeply across the late nineteenth century. A count of smell references that rises with time may therefore be reporting nothing about smell.

Our earlier study [1] detected olfactory references in a subset of the Slovenian Digital Library (dLib.si) covering 1870–1919, using two independent methods: string matching against an expert-curated index of Slovene smell expressions [11], and passage-level extraction with an instruction-tuned LLM. It reported that mentions rise toward the turn of the century and that the two methods agree on only a minority of detections. Both findings were left as observations: the rise was measured in raw counts, and the disagreement was never traced to a cause.

This paper turns those observations into measurements, under a constraint we consider realistic for most heritage projects: **no manual annotation of any kind**. We replace human validation with three automatic corrections and validate the corrections themselves automatically. Our contributions are:

(1) **Sense confirmation**, a label-free method for deciding which entries of a smell-source lexicon are actually being used olfactorily, validated by known-item recovery and split-half reliability;
(2) **an exposure-corrected measurement** of Slovene olfactory discourse showing that its apparent nineteenth-century growth is an artefact of corpus growth;
(3) **a diachronic account of what did change** — the composition and hedonic balance of the printed smellscape — including a result that runs against the standard deodorisation narrative.

2 Related Work

Computational work on historical olfaction is anchored by Odeuropa [9], which produced the European Olfactory Knowledge Graph [6] and an annotation scheme for olfactory events, sources and qualities [7]. That line of work builds gold-standard annotations; we ask what can be established when annotation is unavailable. For Slovene, monolingual resources such as SloBERTa [12] and reference corpora such as Gigafida [5] and JANES [3] support modern text, but not the OCR-degraded, pre-standardised orthography of the period studied here.

Our framing follows the historiography of smell rather than of NLP. Corbin's account of nineteenth-century deodorisation [2] predicts a growing intolerance of odour, and Jenner [4] cautions against reading such narratives off unexamined sources. The corpus-composition problem we address is well documented in culturomics: inferences drawn from raw frequencies in Google Books are strongly confounded by what the corpus contains in each period [10, 8]. To our knowledge this correction has not been applied to sensory mining, and the present study is the first exposure-corrected diachronic measurement of olfactory discourse in Slovene.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.94

3 Data

We use the stored detection output of [1]: for every passage, the list of smell expressions returned by the LLM and the list of matches from the expert keyword index. The archive available to us covers 16,514 documents, 366,333 passages and 109.1 million words published between 1870 and 1919.[1] The keyword method fires in 53.7% of passages and the LLM in 36.4%; 0.68% of passages returned malformed LLM output and are counted as empty.

The exposure problem is visible in the data itself: the corpus grows from 6.3 million words in the 1870s to 41.0 million in the 1910s, a factor of 6.5.

4 Method

All three corrections operate on the existing detections; nothing is re-run against the LLM and nothing is labelled by hand.

4.1 Exposure normalisation

Every quantity we report is a rate per million words with exact Poisson intervals, never a count. Trends are tested with the Mann–Kendall statistic, which assumes neither linearity nor normality. As a negative control we shuffle the year label across passages 200 times and recompute the trend, giving an empirical null for $|\tau|$.

4.2 Sense confirmation

The expert index lists smell *sources* — sweat, earth, wine, incense — and string matching cannot tell whether a given occurrence concerns that thing's smell. In Slovene the problem is acute: *pot* is both 'sweat' and 'path', *duh* both 'odour' and 'spirit', *mira* both 'myrrh' and 'peace'. We therefore ask a second, independent reader. For each term type t we compute

$$c(t) = \frac{|\{\text{passages where the LLM also returns } t\}|}{|\{\text{passages where the keyword index matches } t\}|},$$

after folding diacritics and conservative suffix stripping so that inflected forms of one lemma count as one type, and counting each (passage, type) pair once. Types with at least 20 matches and $c(t) \geq 0.10$ are treated as olfactory; the rest are excluded. The threshold is deliberately low: it is meant to remove types the second reader essentially never corroborates, not to rank the survivors.

Because $c(t)$ is defined without labels, it can be validated without labels. We check (i) *known-item recovery*: whether it ranks a list of explicit Slovene olfactory terms (*vonj*, *smrad*, *dišava*, *zaudarjati*, …), written independently of the score and never shown to it, above the rest of the index; and (ii) *split-half reliability*: whether the score computed on two disjoint random halves of the documents agrees.

4.3 Groundedness

An LLM term that does not occur in its own passage is either an inference or a fabrication. We test each returned term for literal presence after folding and drop those that fail, which bounds the contribution of hallucination without inspecting a single passage by hand.

4.4 Typology

Term types are assigned a domain (food, nature, body, medical, industrial, domestic, ritual), a hedonic value and an explicit/source distinction by a deterministic Slovene lexicon built over the most frequent extracted types. The lexicon labels 50.0% of mention tokens; unlabelled tokens are excluded from composition analyses rather than imputed.

[1]The passages are the same units the 2025 study queried; the archive is a subset of that run, preserving its per-year structure.

Table 1: Automatic validation. No item requires human labels.

Check	Result
Known-item recovery (AUC, explicit terms)	0.92
mean $c(t)$, explicit terms	0.75
mean $c(t)$, other index entries	0.15
Split-half reliability of $c(t)$ (Pearson / Spearman)	0.98 / 0.92
LLM terms present in their own passage	94.8%
Malformed LLM responses	0.68%
Negative control, permuted years ($\lvert\tau\rvert$ at 95th pct.)	0.21
Typology vs. expert domain labels (token-weighted)	77.1%

5 Results

5.1 The corrections

Sense confirmation separates the index sharply. Of 263 term types with at least 20 matches, 119 clear the threshold but carry only 19.2% of keyword matches; the 144 that fail carry 80.5%. Types corroborated in under 2% of their occurrences account for 51.7% of *all* keyword matches in the corpus. The largest are transparently polysemous: *pot* (50,862 matches, $c = 0.001$), *zemlja* 'earth' (25,300, $c = 0.007$), *spomin* 'memory' (14,825, $c = 0.004$) and *svetnik* 'saint' (10,542, $c = 0.0006$), against *smrad* 'stench' ($c = 0.99$) and *vonj* 'scent' ($c = 0.98$). The score is not merely following volume: it recovers the independently written list of explicit olfactory terms with AUC 0.92, and reproduces itself across disjoint halves of the corpus at $r = 0.98$ (Table 1).

Groundedness is high: 94.8% of LLM terms occur literally in their own passage, falling only from 95.4% to 92.8% between the cleanest and noisiest OCR decile. Retaining grounded LLM terms and sense-confirmed keyword matches leaves 355,124 of 723,403 mentions (49.1%).

Two negative findings are worth recording. First, normalising inflection explains almost none of the method disagreement reported in [1]: term-level agreement rises only from 6.9% to 9.0% of detected passages, so 2.2% of the apparent disagreement is a string artefact and the rest is real. Second, the LLM shows no OCR advantage — its passage hit rate falls 42% from the cleanest to the noisiest OCR decile, against 46% for string matching.

5.2 Volume: the rise is the archive

Raw counts reproduce the 2025 result and rise strongly ($\tau = 0.66$, $p < 10^{-10}$), growing 5.5-fold while the corpus grows 6.5-fold. Once expressed per million words the trend vanishes and if anything reverses: $\tau = -0.21$ ($p = 0.03$) over all detections and $\tau = -0.07$ ($p = 0.46$) over the sense-confirmed subset (Fig. 1). The permutation control puts the 95th percentile of $|\tau|$ under shuffled years at 0.21, so even the weak negative trend is at the edge of what year-shuffling produces by chance. The honest statement is that the density of olfactory discourse in Slovene print is *flat* across the fifty years. OCR does not rescue the rise: the junk-token rate has no trend ($p = 0.21$), and conditioning the rate on it leaves the year–rate partial correlation at −0.30.

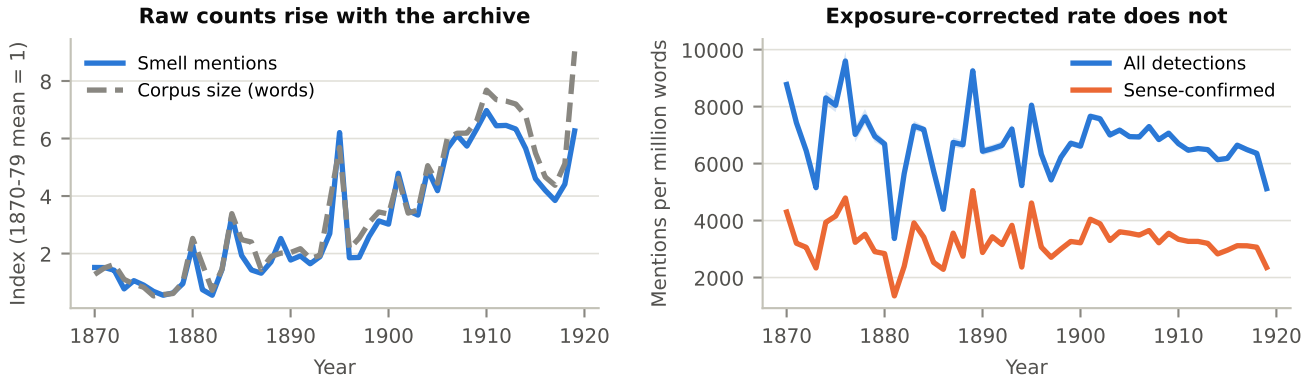


Figure 1: The apparent growth of olfactory discourse is corpus growth. Left: smell mentions and corpus size, each indexed to its 1870–79 mean, move together. Right: expressed as a rate per million words, neither all detections nor the sense-confirmed subset increases.

Table 2: Domain rates per million words, 1870s vs. 1910s, on the sense-confirmed set. Mann–Kendall over all 50 years.

Domain	1870s	1910s	×	p
Food & drink	177.8	326.3	1.84	$< 10^{-5}$
Medicine & hygiene	30.7	50.9	1.66	0.025
Industry & fuel	120.1	183.2	1.53	$< 10^{-5}$
Household	46.7	53.6	1.15	0.027
Ritual	31.0	33.2	1.07	0.99
Body & decay	146.2	115.7	0.79	0.12
Nature & plants	413.2	249.8	0.60	0.040

5.3 Composition: what did change

The mix of smells changes decisively even as the total does not (Fig. 2; $\chi^2(24) = 2023$, $p < 10^{-300}$, Cramér's $V = 0.07$, $n = 110{,}593$). Between the 1870s and the 1910s, food and drink rise from 18.8% to 31.9% of confirmed mentions and industry and fuel from 12.2% to 18.3%, while nature and plants fall from 41.0% to 24.4%. Expressed as rates per million words, food rises 1.8-fold ($\tau = 0.46$, $p < 10^{-5}$) and industry 1.5-fold ($\tau = 0.44$, $p < 10^{-5}$), while nature falls to 0.60 of its initial rate ($\tau = -0.20$, $p = 0.04$) (Table 2). The smellscape of Slovene print moves indoors and into the market, and away from field and forest.

An independent expert labelling of the curated index into these same seven domains (662 terms) reproduces the composition shift and the food, industry and nature results, and agrees with our lexicon on 77.1% of mentions (Table 1). It does not reproduce the medical rise or the flat household row, so we draw no conclusion about either.

5.4 Hedonics: no deodorisation

The deodorisation thesis predicts that smell talk grows more negative. We find the opposite. The share of hedonically marked mentions that are unpleasant falls from 39.3% in the 1870s to 28.5% in the 1910s (52,062 marked mentions; logistic odds ratio 0.93 per decade, $z = -8.2$, $p < 10^{-15}$; Fig. 3).

The drift has a traceable source. A closed list of purchasable scented goods — cigars, spirits, beer, wine, coffee, soap, perfume, ointments, lamp oil — written down before the hedonic result was computed, rises 1.7-fold ($\tau = 0.32$, $p = 0.001$) while everything else falls slightly and non-significantly (0.83-fold, $p = 0.20$). Its share of all confirmed mentions nearly doubles, from 3.7% to 7.1%

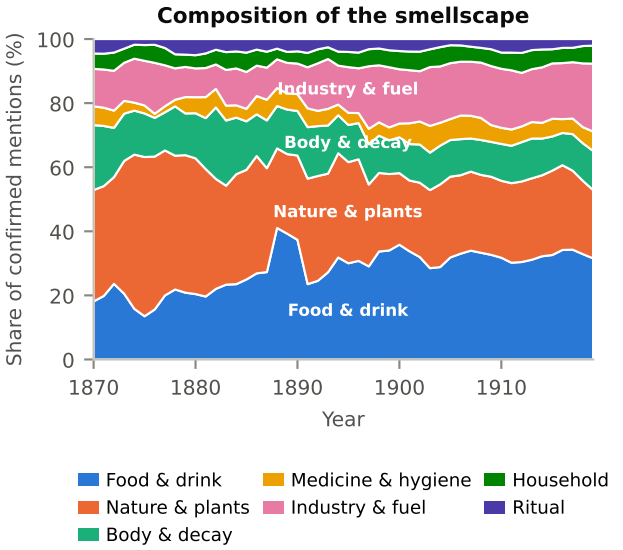


Figure 2: Composition of the sense-confirmed smellscape, three-year rolling mean. The total is flat; the mix is not.

($p < 10^{-4}$). These goods are 20.8% pleasant and 0% unpleasant in our lexicon, against 9.9% and 4.3% for everything else. What our data record is not a culture retreating from odour but a printed public sphere in which an increasing share of smell language is advertising for things that can be bought.

Consistent with this, only 4.1% of confirmed mentions use an explicit olfactory head from our list, and that share declines (odds ratio 0.97 per decade, $p < 10^{-5}$): the corpus talks about smell mostly by naming things that smell. The two methods differ in which things. The keyword index is weighted toward nature (33.4%) and the body (16.9%); the LLM toward food and drink (37.4%) and the household (8.5% vs. 0.5%), and 85.0% of its mentions use vocabulary absent from the expert index entirely.

5.5 Events

We probed three periods with windows fixed from the historical record before the rates were computed. Cholera (1886–87) and the 1895 Ljubljana earthquake give no elevation; the First World War does, with industrial smells at 1.33 and food at 1.30 times

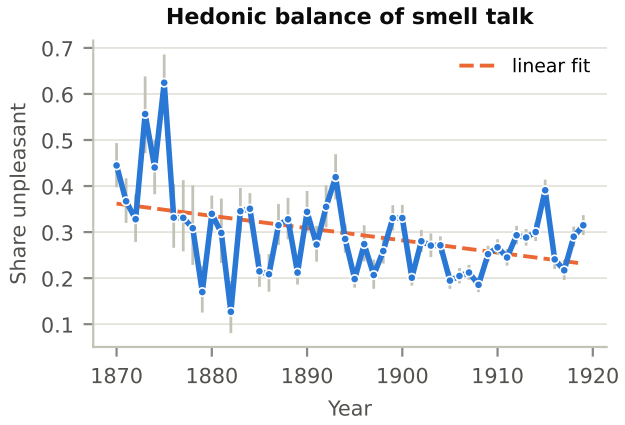


Figure 3: Share of hedonically marked mentions that are unpleasant, with Wilson intervals. The trend is downward, against the deodorisation prediction.

their out-of-window rate ($p = 0.035$ each, Mann–Whitney). Two of three probes are null, which we report as a check on ourselves.

6 Discussion and Limitations

The methodological lesson generalises beyond smell. A curated lexicon of *sources* for any sensory or affective phenomenon will be dominated, in running text, by non-target senses of its most frequent entries; in our corpus half of all matches are such. Where no annotation budget exists, a second detector of a different kind is a usable substitute, not because agreement implies truth, but because systematic non-agreement on a term type is strong evidence of polysemy.

That is also the main limitation. Sense confirmation measures agreement between two detectors, not correctness, and cannot detect an error they share. It is biased toward what the LLM finds legible, and it is a filter, not a precision estimate. We therefore make no absolute claims about detection quality; every claim above is relative or diachronic, where a bias that is constant over time cannot manufacture a change. Valence and domain are assigned at the type level, so an ironic or hostile passage about a perfume still counts as pleasant; the hedonic result is a statement about which vocabulary appears, not about sentiment in context. The available metadata give year but not genre or place of publication, so we cannot separate a change in olfactory culture from a change in the genre mix of what was printed, it is likely that some of both is present. Finally, the corrections use no labelled passages, but domain and valence rest on a word list: ours labels half of all mention tokens, and the expert labelling covers 56.6% of confirmed keyword mentions against only 11.2% of grounded LLM ones, most of which lie outside the curated index.

7 Conclusion

Re-examining 109 million words of Slovene print, we find that the growth of olfactory discourse reported for this period is an artefact of a growing archive: the exposure-corrected rate is flat. What changed is the composition of the smellscape: toward food and industry, away from nature; and its hedonic balance, which becomes more positive rather than more negative, carried by a doubling of the share of purchasable scented goods. Along the way, sense confirmation shows that half of the matches produced by an expert smell lexicon are not about smell at all. We release the per-term table (volume by method, confirmation rate, domain, valence) as an automatically sense-filtered vocabulary of historical Slovene olfactory language, with the expert domain labelling and the pipeline. Future work should recover genre and place of publication from dLib metadata.

Acknowledgements

This work was supported by the Slovenian Research and Innovation Agency via project J7-50233. The smell index is due to Mojca Ramšak [11]. The authors acknowledge the use of large language models for language optimisation. While these tools contributed to re- fining the presentation, all conceptual frameworks, analyses and interpretations remain the sole responsibility of the authors.

References

[1] Janez Brank, Inna Novalija, Dunja Mladenić, and Marko Grobelnik. 2025. LLM based approach to extracting smells in Slovenian corpora. In *Proceedings of the 28th International Multiconference Information Society (SiKDD)*. Ljubljana, Slovenia. doi:10.70314/is.2025.sikdd.5.

[2] Alain Corbin. 1986. *The Foul and the Fragrant: Odor and the French Social Imagination*. Originally published as *Le miasme et la jonquille*, 1982. Harvard University Press, Cambridge, MA.

[3] Darja Fišer, Nikola Ljubešić, and Tomaž Erjavec. 2020. The JANES project: language resources and tools for Slovene user generated content. *Language Resources and Evaluation*, 54, 1, 223–246.

[4] Mark S. R. Jenner. 2011. Follow your nose? smell, smelling, and their histories. *The American Historical Review*, 116, 2, 335–351.

[5] Simon Krek, Špela Arhar Holdt, Tomaž Erjavec, Jaka Čibej, Andraz Repar, Polona Gantar, Nikola Ljubešić, Iztok Kosem, and Kaja Dobrovoljc. 2020. Gigafida 2.0: the reference corpus of written standard Slovene. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*. Marseille, France, 3340–3345.

[6] Pasquale Lisena, Thibault Ehrhart, and Raphaël Troncy. 2024. European olfactory knowledge graph. Zenodo. (2024). doi:10.5281/zenodo.10709703.

[7] Stefano Menini, Teresa Paccosi, Serra Sinem Tekiroğlu, and Sara Tonelli. 2023. Scent mining: extracting olfactory events, smell sources and qualities. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Association for Computational Linguistics, Dubrovnik, Croatia, 135–140. doi:10.18653/v1/2023.latechclfl-1.15.

[8] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331, 6014, 176–182. doi:10.1126/science.1199644.

[9] ODEUROPA Project Consortium. 2021–2023. ODEUROPA: negotiating olfactory and sensory experiences in cultural heritage practice and research. https://odeuropa.eu/. EU Horizon 2020, GA No. 101004469. (2021–2023).

[10] Eitan Adam Pechenick, Christopher M. Danforth, and Peter Sheridan Dodds. 2015. Characterizing the Google Books corpus: strong limits to inferences of socio-cultural and linguistic evolution. *PLoS ONE*, 10, 10, e0137041. doi:10.1371/journal.pone.0137041.

[11] Mojca Ramšak. 2025. *Antropologija vonja*. AMEU-ISH, Ljubljana.

[12] Matej Ulčar and Marko Robnik-Šikonja. 2021. SloBERTa: Slovene monolingual large pretrained masked language model. In *Proceedings of the 24th International Multiconference Information Society (SiKDD)*.

Explaining Headline A/B Test Outcomes with Sparse Autoencoder Features

Drejc Pesjak
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
drejc.pesjak@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Jan Rupnik
Nordoon Inc.
Palo Alto, California, USA

Abstract

A/B testing is a standard method for optimizing news headlines, yet it reveals only *which* variant wins, not *why*. We propose representing headline pairs through Sparse Autoencoder (SAE) features extracted from a large language model. For each within-test pair we construct a *context* vector (shared topic) and a *diff* vector (wording differences), apply histogram-based binarization to suppress activation noise, and train gradient-boosted trees followed by RuleFit to extract interpretable feature patterns. On approximately 57,000 pairs from the Upworthy Research Archive, our best model achieves 61.8% pairwise accuracy and 0.76 AUC. The dominant predictive features are *diff* features with recognizable labels such as "clickbait-style framing," "curiosity gaps," "emotional intensity," and use of the demonstrative "this." We also test a *hierarchy hypothesis*—that topic context gates which diff features matter—but find no stable topic-gated regimes. The results demonstrate that SAE features provide a viable middle ground between opaque dense embeddings and shallow bag-of-words representations for interpretable text analysis.

Keywords

large language models, sparse autoencoders, A/B testing, headline optimization, interpretable machine learning, mechanistic interpretability

1 Introduction

Many decisions in digital media come down to a choice between two variants of the same content, where small differences in wording translate into large differences in reach. A/B testing settles such questions empirically: show both variants to random audiences and measure which one wins. The outcome, however, is merely a winner and a loser. An explanation of why would be more useful, as it carries over to headlines that were never tested.

Moving from winner identification to interpretable explanation requires a text representation that a statistical model can learn from. Early approaches—bag-of-words, TF-IDF, n-grams—are interpretable but miss context and meaning. Dense contextual embeddings from Transformer models [14] such as BERT [6] capture rich semantics but are opaque: no single dimension has a human-readable meaning.

Mechanistic interpretability research offers a promising resolution. Sparse Autoencoders (SAEs) [2, 5] trained on LLM activations decompose dense activation vectors into a much larger, sparse set of features—each corresponding to a recognizable concept such as "shocking scenarios" or "use of the word *this*." These features can be labeled automatically [1], and the GemmaScope-2 project [10] provides pre-trained SAEs for the Gemma model family with human-readable descriptions hosted on Neuronpedia.

In this work we use SAE features as the representation layer for analyzing A/B-tested news headlines from the Upworthy Research Archive [9]. We encode each headline through Gemma-3-4B-IT[1], decompose the activations with a 16,384-width SAE, and construct pairwise *context + diff* vectors that separate shared topic content from wording differences. After binarizing the features to suppress activation noise, we train XGBoost and RuleFit models to identify which SAE features predict the winner and to extract human-readable rules.

We also investigate a hierarchical regime hypothesis: the possibility that the topic or context of an article gates which "difference" features matter for clickability, and test this through unsupervised clustering and model-based recursive partitioning.

2 Related Work

Headline A/B testing and engagement. The Upworthy Research Archive [9] made a large corpus of real headline A/B tests publicly available for research. Gligorić et al. [8] analyzed the same archive with logistic regression on hand-crafted linguistic features (word count, sentiment, named entities, syntactic patterns), finding small but significant effects of specific wording choices on click-through rates. Robertson et al. [13] used randomized Upworthy trials to show that negatively worded headlines drive higher consumption. Clickbait detection work [3] has identified surface-level engagement cues such as forward references and sensational language. Our approach differs from all of these in using SAE features, which provide richer, model-derived interpretable dimensions rather than manually defined linguistic variables or opaque dense vectors.

Sparse autoencoders and interpretable text representations. Sparse autoencoders trained on neural network activations learn overcomplete dictionaries where individual features correspond to human-recognizable concepts [2, 5]. The GemmaScope project [10] provides pre-trained SAEs for the Gemma model family, including JumpReLU variants [12] that use a learnable threshold to enforce sparsity. Feature labels can be generated automatically by having an LLM describe each feature's activation pattern [1]. Most directly related to our work, Movva et al. [11] train SAEs on sentence embeddings and use L1-penalized logistic regression

[1] google/gemma-3-4b-it on Hugging Face.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.89

to select predictive features for hypothesis generation. Our work differs in three ways: (i) we use residual-stream SAEs rather than embedding-space SAEs, (ii) we introduce a context+diff decomposition for pairwise comparisons, and (iii) we use tree-based models and RuleFit for conjunctive rule extraction rather than linear feature selection.

Interpretable machine learning. RuleFit [7] combines tree-derived conjunction rules with linear terms, producing models that are both predictive and human-readable. XGBoost [4] provides strong gradient-boosted baselines with built-in feature importance. Model-based recursive partitioning (MOB) [15] tests for parameter instability across subgroups, offering a principled way to discover heterogeneous treatment effects—we use it to test for topic-gated regime structure.

3 Methods

3.1 Dataset

We use the Upworthy Research Archive [9], which contains A/B test data from the news website Upworthy. Each "test" compared 2–5 headline variants for the same article, with real impression and click counts used to calculate click-through rate (CTR). We use the *confirmatory* split, comprising approximately 13,500 unique test IDs.

After preprocessing (HTML tag removal, deduplication, removal of single-headline tests), we sample within-test headline pairs. For each test with N headlines, we draw up to $\min\left(\binom{N}{2}, 10\right)$ random pairs, yielding approximately 57,289 pairs with balanced labels. The pair order is randomized, and the binary label is:

$$y = \mathbf{1}\left[\mathrm{CTR}_A > \mathrm{CTR}_B\right] \tag{1}$$

where CTR = clicks/impressions for each headline. We split into training (45,657 pairs) and test (11,632 pairs) sets using `GroupShuffleSplit` by `test_id`, ensuring that all pairs from the same test remain in the same split to prevent data leakage.

3.2 SAE Feature Extraction

Each headline is wrapped in a Gemma-3 chat template and passed through Gemma-3-4B-IT (Google's 4-billion parameter instruction-tuned model). We extract residual stream activations $\mathbf{h}_t \in \mathbb{R}^{2560}$ at layer 22 for each token position t, masking out instruction template tokens so that only headline content tokens remain.

The content-token activations are then encoded through a pre-trained JumpReLU SAE from the GemmaScope-2 project [10]:

$$\mathbf{z}_t = \mathrm{JumpReLU}_\theta\left(\mathbf{W}_{\mathrm{enc}}\,\mathbf{h}_t + \mathbf{b}_{\mathrm{enc}}\right) \tag{2}$$

where $\mathbf{W}_{\mathrm{enc}} \in \mathbb{R}^{d_{\mathrm{SAE}} \times d_{\mathrm{model}}}$, $d_{\mathrm{SAE}} = 16{,}384$, and the JumpReLU activation sets entries below a learned threshold θ to zero:

$$\mathrm{JumpReLU}_\theta(x) = x \cdot \mathbf{1}[x > \theta] \tag{3}$$

To obtain a single vector per headline, we max-pool across all content token positions:

$$\mathbf{v} = \max_{t \in \mathcal{T}_{\mathrm{content}}} \mathbf{z}_t \tag{4}$$

yielding a sparse vector $\mathbf{v} \in \mathbb{R}^{16384}_{\geq 0}$ where each dimension corresponds to a named SAE feature (e.g., "discovery and revelation," "shocking scenarios").

3.3 Pairwise Context + Diff Representation

Predicting CTR from a single headline's SAE vector would mostly capture which *topics* are inherently popular, not what makes one headline beat another for the same article. Since we have within-test pairs, we can focus on wording differences.

For a pair (A, B) with SAE vectors $\mathbf{v}_A$ and $\mathbf{v}_B$, we compute:

$$\mathbf{c} = \frac{\mathbf{v}_A + \mathbf{v}_B}{2} \qquad \text{(context: shared topic/content)} \tag{5}$$

$$\mathbf{d} = \mathbf{v}_A - \mathbf{v}_B \qquad \text{(diff: wording differences)} \tag{6}$$

and concatenate them into a single feature vector:

$$\mathbf{x} = [\,\mathbf{c} \parallel \mathbf{d}\,] \in \mathbb{R}^{32768} \tag{7}$$

The diff vector encodes *wording* (not just word) *differences* between the two variants in the SAE feature basis: it captures whatever changes in the model's representation when the phrasing is altered, from surface cues (e.g., demonstratives, punctuation, syntactic templates) to higher-level semantic or stylistic framing (e.g., emotion, moral framing, curiosity gaps). The context vector captures the shared topic, included to allow models to potentially learn context-dependent rules (e.g., "emotional language helps for political articles but not for lifestyle content").

Each pair is weighted by the absolute CTR difference to emphasize pairs with clearer outcomes:

$$w = \left|\mathrm{CTR}_A - \mathrm{CTR}_B\right| \tag{8}$$

3.4 Feature Binarization

Raw SAE activations contain noise near the JumpReLU cutoff. The non-zero values exhibit characteristic distributional structure (Figure 1) that we exploit for denoising.

For *context* features (the per-pair mean activation) the distribution has two overlapping components. When only one headline of the pair fires the feature the mean is half the single-headline activation, producing a steeply decaying exponential just above the JumpReLU cutoff. When both headlines fire, the mean of two draws yields a bell-shaped body at higher values. For *diff* features the distribution has a central Gaussian at zero (both headlines activate the feature similarly), flanked by exponential-like tails shifted into the positive and negative directions (one headline activates the feature substantially more than the other).

We separate noise from signal with a histogram-based "walk" threshold. For context, we walk rightward from the smallest non-zero bin: when 4 consecutive bins are all higher than the current bin, the transition from the decaying component to the bell-shaped body has been reached, yielding threshold $\tau_{c,j}$ and binary encoding $\tilde{c}_j = \mathbf{1}[c_j > \tau_{c,j}]$. For diff features the same walk proceeds outward from zero in both directions, yielding symmetric thresholds $\pm\tau_{d,j}$ and trinary encoding:

$$\tilde{d}_j = \begin{cases} +1 & \text{if } d_j > +\tau_{d,j} \\ -1 & \text{if } d_j < -\tau_{d,j} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

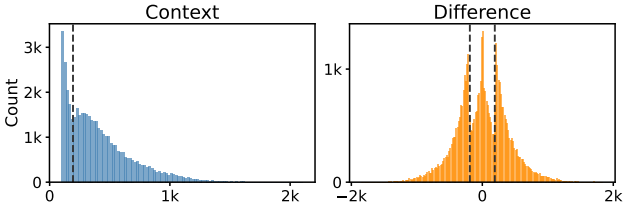


Figure 1: Context (left) and difference (right) activations for SAE feature 604 ("harmful unethical behavior"). Dashed lines mark walk thresholds.

3.5 Models

We evaluate the following models, all on the same group-based train/test split:

Logistic Regression. SGD-trained logistic regression with L2 regularization ($\alpha = 10^{-4}$) on the full 32,768-dimensional continuous feature vectors (we apply StandardScaler without centering to preserve sparsity), serving as a linear baseline.

XGBoost on binarized features. Gradient-boosted trees [4] with 100 estimators, maximum depth 6, learning rate 0.1, and column subsampling of 0.7, trained on the binarized/trinarized 32,768 int8 features. Sample weights are set to w (Equation 8). Only 467 out of 32,768 features receive non-zero importance from this model.

RuleFit on XGBoost-selected features. We extract the 467 features with non-zero XGBoost importance and train a RuleFit model [7] (25 estimators, tree size 4, maximum 500 rules). RuleFit generates both linear terms (each binarized feature with a coefficient) and conjunction rules (tree-derived feature combinations), producing a fully interpretable model.

Additional baselines. We also train a depth-8 decision tree (for visual interpretability), XGBoost with depth-1 stumps (to test whether additive features suffice), and per-cluster XGBoost models (to test the hierarchy hypothesis).

4 Results

4.1 Main Results

Experimental evaluation compares all models on the same group-based train/test split with sample weights $w = |\mathrm{CTR}_A - \mathrm{CTR}_B|$. Table 1 summarizes the results giving three commonly used evaluation metrics: classification accuracy (Acc), harmonic mean of precision and recall (F1) and area under the Receiver Operating Characteristic curve (AUC).

Table 1: Performance of all evaluated models on the held-out test set.

#	Method	Acc	F1	AUC
1	Logistic Regression	0.606	0.606	0.750
2	Decision Tree	0.556	0.551	0.630
3	XGBoost (continuous)	0.616	0.616	0.759
4	XGBoost per-cluster	0.571	0.564	0.673
5	**XGBoost (binarized)**	**0.618**	**0.619**	0.759
6	XGBoost stumps (d=1)	0.577	0.578	0.681
7	Decision Tree (binarized)	0.553	0.549	0.625
8	RuleFit (467 features)	0.588	0.556	**0.761**

Logistic regression is competitive (AUC 0.750), while XGBoost performs slightly better (0.759), indicating useful but limited nonlinear interactions. Depth-6 trees outperform stumps (0.759 vs. 0.681 AUC). Binarization slightly improves accuracy (61.6% → 61.8%) and concentrates importance on 467 of 32,768 features. RuleFit reaches comparable AUC (0.761) with 52 conjunction rules and 443 linear terms.

4.2 Top Predictive Features

Table 2 shows the ten most important linear terms from the RuleFit model. All are *diff* features—the SAE dimensions that capture how two headlines differ—confirming that *what distinguishes the headlines* matters far more than *what the article is about*. Context features appear only rarely among the top importances.

Table 2: Top 10 RuleFit linear terms by importance. All are diff features. A positive coefficient means that headline A activating this feature more than headline B predicts that A wins (higher CTR).

Rank	SAE Feature Label	Coef	Imp
1	"this" (demonstrative)	+0.14	0.076
2	clickbait-style titles	+0.11	0.070
3	remarkable, amazing qualities	+0.11	0.068
4	discovery and revelation	+0.10	0.061
5	subtitles	−0.12	0.061
6	past actions or events	+0.11	0.059
7	shocking or horrifying scenarios	+0.10	0.057
8	harmful unethical behavior	+0.08	0.056
9	math and trick questions	+0.09	0.056
10	stereotypes about gender roles	+0.09	0.055

Table 3: Examples of interpretable conjunction rules extracted by RuleFit. Positive and negative diff values favor headline A and B, respectively.

#	Rule	Coef
1	`clickbait titles` > 0.5	+0.07
2	`unexpected things` > 0.5	+0.06
3	`intense negative emotions` ≤ -0.5	−0.05
4	`recognizing or noticing` ≤ -0.5	−0.14
5	`joke, humor` > -0.5 and `things or anything` ≤ 0.5 and ctx: `asking questions` > 0.5	+0.03
6	`regulation, risk` > 0.5 and `past events` > -0.5	−0.04

The features tell a coherent story: headlines that use the demonstrative "this" (creating a curiosity gap), employ clickbait-style framing, evoke emotional intensity (shock, outrage), or promise revelation tend to get more clicks. Dry, subtitle-style text (feature 5, negative coefficient) hurts clickability.

4.3 Example Interpretable Rules

Table 3 shows selected RuleFit conjunction rules capturing interactions among binarized SAE features.

For instance, the fifth rule states that in the context of question-framed articles (context feature active), if the diff in humor is not strongly negative and the diff in "things or anything" vagueness is low, the more humorous variant tends to win. The sixth rule shows that regulatory/governance language combined with past-event framing *hurts* clickability.

Illustrative headline pair. Consider a real A/B test from the Upworthy archive where variant A is *"The Secret About Work Life In 20 Other Countries That Will Surprise You"* and variant B is *"In Japan, They Have A Word For Working Yourself To Death."* Both headlines cover the same topic (international work culture), so their context vectors are similar. However, variant A strongly activates the diff features for "clickbait-style titles" (+1) and "discovery and revelation" (+1) through its "The Secret…That Will Surprise You" framing, while variant B is specific and informative but does not trigger these engagement features. The model predicts A as the winner, consistent with the observed A/B test outcome.

4.4 Hierarchy Hypothesis

A central question was whether *context-dependent rules* govern clickability—for example, whether emotional language helps for political headlines but hurts for lifestyle content. We tested this through three approaches:

K-Means clustering on SVD-reduced context vectors (10 clusters) followed by per-cluster XGBoost models. The aggregated per-cluster predictions achieved 57.1% accuracy (vs. 61.8% global), a 4.7 percentage point degradation. The clusters fragmented the data without capturing distinct clickability regimes.

Model-Based Recursive Partitioning (MOB) [15] using context features as splitting variables and per-leaf Lasso regression on diff features. The MOB tree found statistically significant parameter instability (via the supLM test), producing 10 leaves with different regression coefficients. However, performance was severely overfit: train $R^2 = 0.39$ vs. test $R^2 = 0.04$, because each leaf had too few samples relative to the 16,384 diff features.

XGBoost tree dump analysis. We audited the split variables in the depth-6 binarized XGBoost model. Of 1,553 total splits, only 188 were on context features (12.1%), with the remaining 1,365 on diff features. Context splits appeared primarily near leaves, not at the root—suggesting the ensemble does not implement "context-first gating" in any strong sense.

These results indicate that while non-additive interactions exist (depth-6 substantially outperforms depth-1 stumps), they are predominantly *among diff features*—not between context and diff. The hierarchy hypothesis, as tested, is not supported.

5 Discussion and Conclusions

Our experimental analysis has shown that on our data the best model achieves 61.8% accuracy and 0.76 AUC. This is a modest but real signal, given that many A/B test pairs have very small CTR differences, which makes the ground-truth label itself noisy. The SAE features come from a general-purpose LLM that was never fine-tuned for headline effectiveness, yet they capture recognizable engagement patterns. Across all models the diff features dominate, so what distinguishes two headlines matters far more than what the article is about. The top features (clickbait framing, curiosity gaps, emotional intensity, use of "this") align with known headline optimization principles and provide independent empirical validation. Depth-6 trees outperform depth-1 stumps by 7.8 AUC points, suggesting that multi-way feature interactions carry signal, yet our attempts to find stable topic-gated regimes were unsuccessful. The interactions are among the diff features themselves rather than between context and diff, an informative negative result suggesting that hierarchical structure, if present, does not follow the topic dimension we assumed.

We conclude that Sparse Autoencoder features extracted from a large language model provide an interpretable representation for analyzing headline A/B test outcomes. The context + diff decomposition, combined with histogram-based binarization, produces human-readable feature importances and conjunctive rules at predictive performance comparable to continuous representations, placing SAE features between opaque dense embeddings and shallow bag-of-words representations.

Limitations and future work. The multi-stage pipeline compounds approximation errors, and the SAE feature labels are generated automatically, so they may not perfectly describe what each feature detects. All reported associations are predictive rather than causal: without controlled experiments we cannot claim that activating a particular SAE feature causes more clicks. Hard partitioning methods (k-means, MOB) may also miss soft or overlapping regime structure. Future work could explore larger SAE dictionaries (65k or 262k width), models fine-tuned for headline tasks, causal validation through controlled generation, and soft mixture-of-experts approaches to regime discovery. Future work should compare SAE features against lexical and dense representations and isolate the contributions of context and binarization through controlled ablations.

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency (ARIS) through the Young Researcher Grant (PR-13856), the research program Knowledge Technologies (P2-0103) and the Gravity project AI for Science (GC-0001) and European HE project DataPact (GA 101189771).

The authors acknowledge the use of Claude to support paper organization and language and grammar refinement. All AI-generated suggestions were carefully reviewed and revised by the authors, who take full responsibility for the content of this paper.

References

[1] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Joshua Clymer, et al. 2023. Language models can explain neurons in language models. *OpenAI blog*.

[2] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Henighan, Tom Conerly, et al. 2023. Towards monosemanticity: decomposing language models with dictionary learning. *Transformer Circuits Thread*. Anthropic.

[3] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: detecting and preventing clickbaits in online news media. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 9–16.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

[5] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 4171–4186.

[7] Jerome H. Friedman and Bogdan E. Popescu. 2008. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2, 3, 916–954.

[8] Kristina Gligorić, Ashton Anderson, and Robert West. 2023. Linguistic effects on news headline success: evidence from thousands of online field experiments (registered report). *PLOS ONE*, 18, 3, e0257091. doi:10.1371/journal.pone.0257091.

[9] J. Nathan Matias, Kevin Munger, Marianne Aubin Le Quéré, and Charles Ebersole. 2021. The Upworthy Research Archive: a time series of 32,488 experiments in U.S. media. *Journal of Quantitative Description: Digital Media*, 1. Dataset: https://osf.io/jd64p/. doi:10.51685/jqd.2021.014.

[10] Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. 2025. Gemma scope 2: technical paper. https://huggingface.co/google/gemma-scope-2.

[11] Rajiv Movva, Kenny Peng, Nikhil Garg, Jon M. Kleinberg, and Emma Pierson. 2025. Sparse autoencoders for hypothesis generation. In *Forty-second International Conference on Machine Learning (ICML)* (Proceedings of Machine Learning Research). Vol. 267. PMLR. https://proceedings.mlr.press/v267/movva25a.html.

[12] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Jumping ahead: improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*.

[13] Claire E. Robertson, Nicolas Pröllochs, Kaoru Schwarzenegger, Philip Pärnamets, Jay J. Van Bavel, and Stefan Feuerriegel. 2023. Negativity drives online news consumption. *Nature Human Behaviour*, 7, 5, 812–822. doi:10.1038/s41562-023-01538-4.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 5998–6008.

[15] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. 2008. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17, 2, 492–514.

Same Labels, Smaller Bill: An Ablation Study of LLM Claim Classification in Cross-Source News

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan Postgraduate School
Ljubljana, Slovenia

Blaž Škrlj
blaz.skrlj@ijs.si
Jožef Stefan Institute
Jožef Stefan Postgraduate School
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Jožef Stefan Postgraduate School
Ljubljana, Slovenia

Abstract

Comparing how news outlets cover the same event requires labelling each claim per outlet; we ask which LLM set-up does this accurately and cheaply. Our prior multi-stage prompt architecture reduced classification errors 7-fold but was evaluated with a single model (gpt-4o-mini), prompt and decoding setting. Here we evaluate 30 configurations in 78 runs, spanning two classification architectures, eight LLMs from five providers (OpenAI, Google, Anthropic, xAI, DeepSeek), prompt variants for both architectures and a temperature sweep, all with reasoning off, minimal or low (grok-4.1-fast at its default reasoning). Using 1,079 manually annotated claim–publisher pairs from 10 Slovenian news events, we report exact and relaxed accuracy, per-class F1 and cost, and compare against a non-LLM sentence-embedding baseline. The best configurations reach 82% exact and 93–95% relaxed accuracy (gemini-3-flash and gemini-2.5-flash-lite with the reduced per-publisher prompt, gpt-5-nano with the label_only_no_rules per-claim prompt); the best budget-tier configurations match or exceed the tested frontier configurations with reasoning off or at low effort; the per-publisher architecture is 2.5–5.2× cheaper at equal or higher accuracy; and every LLM configuration clearly outperforms the embedding baseline (57.5%).

Keywords

News Analysis, Stance Detection, Claim Verification

1 Introduction

Comparing how publishers cover the same event is essential for understanding media bias and information diversity; we ask which LLM set-up does it best at low cost. In our prior work [1], we presented BetweenTheLines, a system that extracts claims from news articles and classifies whether each publisher confirms, partially confirms, contradicts, or omits each claim. The key finding was that a multi-stage LLM architecture (separate extraction and classification prompts) reduced the rate of reviewer-flagged errors from 33.0% to 4.67% over 797 claim–publisher pairs, compared with a single prompt that reads all articles at once.

Those experiments, however, used a fixed configuration (gpt-4o-mini as the sole model, a single detailed prompt, default temperature), compared the multi-stage design only with a single prompt rather than with the two ways its own classification stage can be organized, and had no non-LLM baseline. How far the results generalize is therefore open, and this study asks five research questions:

RQ1 (Architecture): Does the classification architecture (per_publisher vs per_claim_publisher, Figure 1) affect accuracy and cost?

RQ2 (Models): Do different LLM providers and models perform differently on this task?

RQ3 (Prompts): Do prompt length and explicitly stated classification rules affect accuracy?

RQ4 (Temperature): Does temperature affect classification consistency and accuracy?

RQ5 (Baseline): Can non-LLM baselines compete with LLMs for claim classification?

We contribute a systematic evaluation of 30 configurations in 78 runs across architectures, models, prompts and temperature; an annotated dataset of 1,079 claim–publisher pairs from 10 events; a non-LLM embedding baseline; and practical cost–accuracy guidance. The dataset, evaluation scripts and full per-class metrics are public.[1]

2 Related Work

Our work intersects claim verification and stance detection; Dmonte et al. [2] review LLM-based claim verification, including chain-of-thought prompting and retrieval-augmented methods. Automated fact-checking asks whether a claim is true or supported by evidence [3]; claim comparison asks instead whether each outlet includes, omits or presents an equivalent claim differently, with the news event as the unit of comparison [4].

LLMs exhibit notable sensitivity to prompt formulation. Errica et al. [5] show that minor wording changes produce significant performance variation. Regarding decoding settings, Renze and Guven [6] find that sampling temperature between 0 and 1 has no statistically significant effect on multiple-choice problem-solving accuracy across nine LLMs.

[1] https://github.com/GeorgiGeor/datamining-ablations-newsdiff-data

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SiKDD 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.113

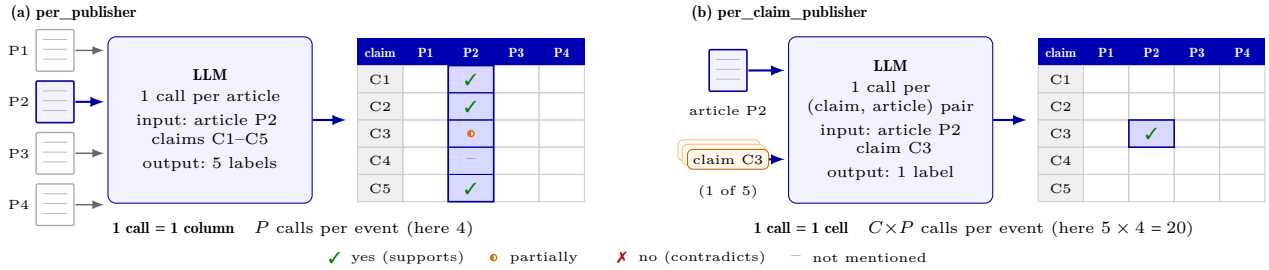


Figure 1: Illustrative classification architectures on a hypothetical event with 4 articles and 5 claims: (a) `per_publisher`, one call per article for all claims; (b) `per_claim_publisher`, one call per (claim, article) pair.

These findings motivate our comparison of prompt variants (RQ3) and of whether the temperature result transfers to our classification task (RQ4). Intermediate labels are hard elsewhere too: in natural language inference the *neutral* class has the lowest annotator agreement [7].

3 Methodology

3.1 Task and Label Definitions

Claim. A claim is a short, self-contained declarative statement drawn from an event's coverage—a factual assertion, an accusation, or a statement attributed to an actor—that can be checked against the text of an individual article. Claims are defined at the *event* level: the extraction stage reads all articles of an event jointly and returns 8–15 such statements, favoring those that are central to the story or reported by several outlets; opinions of the reporting outlet are not claims. For example, one of the events in our dataset (Section 3.2) concerns the EU investigation of the platform X; one of its extracted claims reads "X was already fined 120 million euros in December for transparency violations".

Claim–publisher pair and labels. The unit of evaluation is the pair (claim, article). Its label states whether the article *asserts or supports* the claim (`yes`), *mentions it only indirectly*—in paraphrase, in part, or in a broader or narrower form (`partially`), *contradicts or denies* it (`no`), or *does not address it at all* (`not_mentioned`). A claim may thus be absent from most articles and still be a valid claim; the example above is labelled `yes` for seven of the event's nine publishers and `not_mentioned` for two. A single annotator labelled all 1,079 pairs; the distribution is 52.8% `yes`, 37.5% `not_mentioned`, 9.5% `partially` and 0.1% `no` (n=1). The `no` class is rare because news articles rarely contradict each other directly.

3.2 Dataset

We constructed an evaluation dataset from Slovenian news coverage collected with Event Registry [8], a news aggregator that clusters articles from different outlets into *events* (one news story each): 10 events from January 2026, 87 articles, 124 distinct claims and 1,079 annotated claim–publisher pairs. Claims were extracted once per event with gemini-3-flash-preview from all articles concatenated (about 6,800 input tokens per call); the resulting claim list was annotated and is shared by every configuration, which isolates classification from extraction variability.

The events span diverse topics to test generalization: EU/technology regulation, sports, domestic politics, international relations, and economic policy. Each event contains between 7–10 articles from different Slovenian publishers and 12–13 claims. We denote by P the number of articles (publishers) and by C the number of claims of an event.

3.3 Classification Architectures

Classification is the second, ablated stage: it receives an event's fixed claim list and its articles and labels every (claim, article) pair. Figure 1 contrasts the two ways of doing so.

`per_publisher`: One LLM call per article classifies all C claims simultaneously. Minimizes API calls and cost (P calls per event).

`per_claim_publisher`: One LLM call per (claim, article) pair. More granular but requires $C \times P$ calls per event instead of P.

A per_publisher call averages about 1,800 input tokens, a per_claim_publisher call 1,100.

3.4 Models

We evaluate eight LLMs across five providers, organized by the providers' own tiering—flagship, mid-size, small (Table 1), plus a non-LLM embedding baseline. Hereafter we abbreviate gemini-3-flash-preview as gemini-3-flash.

Tier	Model	Provider	Reasoning	T
Frontier	gpt-5.2	OpenAI	low	n/a
Frontier	deepseek-v3.2	DeepSeek	off	0.3
Mid	gpt-5-mini	OpenAI	low	n/a
Mid	claude-sonnet-4.5	Anthropic	off	0.3
Mid	grok-4.1-fast	xAI	on	1
Mid	gemini-3-flash-preview	Google	minimal	1
Budget	gpt-5-nano	OpenAI	low	n/a
Budget	gemini-2.5-flash-lite	Google	off	0/1/2
Baseline	embedding	Local	—	—

Table 1: Models, reasoning setting and temperature (T).

Reasoning effort. Reasoning was left off (provider default) for Claude, DeepSeek and gemini-2.5-flash-lite, set to `minimal` for gemini-3-flash and to `low` for the GPT-5 family (Table 1); the exception is grok-4.1-fast, whose provider default left reasoning enabled. It is a fixed condition of this study, not an ablated factor: all findings hold with reasoning off, minimal or low (Section 5).

Embedding baseline. We implement a non-LLM baseline using sentence-transformers [9] (model `paraphrase-multilingual-MiniLM-L12-v2`) with cosine-similarity thresholds: $>0.55 \rightarrow$ `yes`, 0.35–0.55 $\rightarrow$ `partially`, $\leq 0.35 \rightarrow$ `not_mentioned`. This baseline cannot distinguish `yes` from `no` since embeddings capture relatedness, not stance. Thresholds were fixed a priori and not tuned on the evaluation set.

3.5 Prompt Variants

We test prompt variants appropriate to each architecture. For **per_publisher**: **baseline** (label definitions, a mandatory checklist of transformations—paraphrase, hypernyms, abbreviations, coreference—to check before answering `not_mentioned`, and a tie-break preferring `partially`) and **minimal** (reduced instructions). For **per_claim_publisher**: **baseline** (the same guidance for a single (claim, article) pair), **shortened** (condensed baseline), **label_only_rules** (label-only output with rule guidance), and **label_only_no_rules** (question and labels only). All prompts written in Slovenian to match article language.

3.6 Temperature and Runs

A *configuration* is a combination of architecture, model, prompt variant and temperature; a *run* is one pass of a configuration over all 1,079 pairs. Temperature was swept at 0, 1 and 2 for gemini-2.5-flash-lite only (nine configurations: three temperatures for each of three architecture/prompt cells; Google recommends T=1 for Gemini). All other models ran at a single value (Table 1): 0.3 for Claude and DeepSeek, 1 for Grok and Gemini 3; the GPT-5 family has no temperature parameter. Every configuration was run on the same claims and articles one to three times (Table 2): the T=0 configurations, grok-4.1-fast (retired by its provider during the study) and the embedding baseline once, all other configurations three times. Runs with more than 1% unlabelled pairs were discarded, and two January runs whose exported predictions were partly substituted by embedding-baseline output (an export defect—the underlying API runs were clean) were excluded and replaced by their configurations' next runs. In total, 30 configurations and 78 runs.

3.7 Evaluation Metrics

Exact accuracy: Percentage matching ground truth exactly. **Relaxed accuracy:** Counts adjacent label confusions as acceptable—specifically `yes`↔`partially` and `partially`↔`not_mentioned`. The `no` class requires exact match. **Cost:** USD per run (10 events), computed from the logged token counts of every call and provider list prices; the embedding baseline makes no API calls. All accuracies are pooled over the 1,079 pairs (the mean over events differs by at most 1 pp); configuration-level results are means over a configuration's runs. Exact accuracy on four labels is stricter than the reviewer-judged error rate of the 2025 evaluation; relaxed accuracy is its closer counterpart. **Uncertainty:** The 1,079 pairs are clustered within 10 events and are not independent; a bootstrap that resamples whole events (10,000 resamples) puts the 95% confidence interval of a run's accuracy at about ±6 pp in the median (±4–7 pp at the accuracy levels of Table 2). Configurations are however compared on the same pairs, so we judge differences against run-to-run spread: the standard deviation of a configuration is typically below 1 pp (Table 2) and we read differences of about 2 pp or less as within run-to-run variation. Exact McNemar tests between single runs are provided in the repository.

4 Results

The 78 runs cost $24.5 in total. Table 2 lists the ten best configurations ordered by mean exact accuracy, plus the embedding baseline; they span 2.4 pp, well within the event-level uncertainty (Section 3.7) and close to the run-to-run spread, so their order should not be over-interpreted.

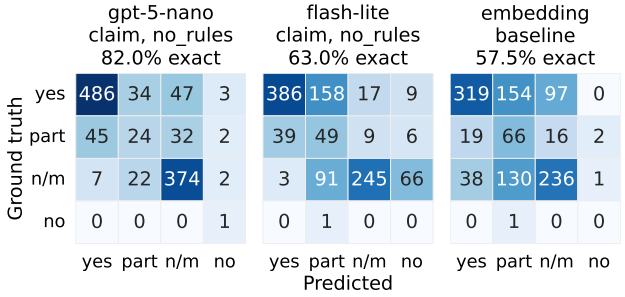


Figure 2: Confusion matrices of three single runs (not configuration means).

#	Model	Arch	Prompt	T	Exact	Relax	Cost	n
1	gemini-3-flash	pub	min	1	82.4±0.2	94.3	$0.21	3
2	gemini-2.5-flash-lite	pub	min	0	81.8	94.6	$0.03	1
3	gpt-5-nano	claim	no_rules	–	81.8±0.6	93.3	$0.14	3
4	gemini-2.5-flash-lite	pub	min	1	81.0±0.2	94.1	$0.03	3
5	gemini-2.5-flash-lite	pub	min	2	80.9±0.6	94.3	$0.03	3
6	gpt-5-mini	pub	base	–	80.5±0.1	96.4	$0.28	3
7	claude-sonnet-4.5	pub	min	0.3	80.4±0.2	95.0	$1.10	3
8	gemini-3-flash	pub	base	1	80.2±0.1	95.9	$0.28	3
9	gemini-2.5-flash-lite	pub	base	1	80.1±0.7	96.0	$0.05	3
10	gpt-5-mini	claim	base	–	80.0±0.4	96.8	$0.70	3
30	embedding	—	—	–	57.5	87.1	—	1

Table 2: The ten best of the 30 configurations by mean exact accuracy (±sd over the n runs); no_rules = label_only_no_rules; cost per run.

RQ1 (Architecture): per_publisher fills eight of the ten places in Table 2, including the two best configurations, and gpt-5-nano (81.8%) and grok-4.1-fast (79.5%) are the only models whose best result comes from per_claim_publisher. Within the same model and prompt, per_publisher is more accurate by 6.6 pp for gemini-2.5-flash-lite (80.1% vs 73.5%), 4.6 pp for gemini-3-flash and 2.8 pp for deepseek-v3.2—well beyond run-to-run variation—and within 1.4 pp for gpt-5-nano, gpt-5-mini and grok. Since per_publisher is also 2.5–5.2× cheaper for the same model (2.5× for gpt-5-mini, 5.2× for deepseek-v3.2) because of its far fewer API calls (Figure 1), it is the better default; the GPT-5 models and grok lose little or nothing when each claim is judged in isolation.

RQ2 (Models): The best observed budget-tier configurations match or exceed the tested frontier configurations with reasoning off or at low effort. The best configuration of each tier reaches 82.4% for Mid (gemini-3-flash, per_publisher minimal), 81.8% for Budget (gemini-2.5-flash-lite, per_publisher minimal, T=0, and gpt-5-nano) and 79.1% for Frontier (gpt-5.2). gpt-5.2 is thus 3.2 pp below the best configuration and 2.6 pp below gpt-5-nano label_only_no_rules, and level with gemini-2.5-flash-lite per_publisher (79.1% vs 80.1%); deepseek-v3.2 (75.9%) is 4.2 pp below the latter. The best cost–accuracy point is gemini-2.5-flash-lite per_publisher with the minimal prompt at $0.03 per run for 81.8%.

RQ3 (Prompts): Both architectures have prompt variants (Section 3.5). For per_claim_publisher, verbosity has no consistent effect but the classification rules do: for gpt-5-nano the `label_only_no_rules` prompt scores 4.2 pp above baseline (81.8% vs 77.5%), suggesting reasoning models do

well with less prescriptive instructions, while `shortened` gains 2.1 pp; for gemini-2.5-flash-lite, `label_only_rules` (78.3%) and `shortened` (77.2%) beat baseline (73.5%) by 3.7–4.8 pp, but `label_only_no_rules` collapses to 64.0%. Keeping the rule checklist matters more than verbosity, and how much depends on the model. For per_publisher the reverse holds: the minimal prompt scores 0.9–2.2 pp above the baseline mean for every Gemini configuration and 0.7 pp for Claude—at the edge of run-to-run variation—whereas for gpt-5-nano it is 2.2 pp worse (76.7% vs 78.9%).

RQ4 (Temperature): No detectable effect for gemini-2.5-flash-lite, the only model swept. With per_publisher and the baseline prompt, the T=0 run (79.6%) lies within the spread of the six runs at T≥1 (76.5–80.6%, mean 79.5%, sd 1.5); the T=1 and T=2 means differ by 1.1 pp in one direction for per_publisher (80.1% vs 79.0%) and, for per_claim_publisher, by 1.1 pp in the other (73.5% vs 74.6%), and the minimal prompt gives 81.8%, 81.0% and 80.9% at T=0, 1 and 2.

RQ5 (Baseline): LLMs substantially outperform embedding similarity. The LLM configurations reach 82% exact accuracy vs 57.5% for embeddings (+24.8 pp for the best configuration). The gap is largest for F1-not_mentioned (0.89 vs 0.63), where embeddings struggle to distinguish semantic absence from low similarity.

Per-class performance: F1-no is not meaningful because of the single `no` sample, which 35 of the 78 runs label correctly. The `partially` class remains the hardest (best F1 = 0.41 in any run), consistent with the neutral-class difficulty noted in Section 2, while `yes` and `not_mentioned` reach F1 > 0.84 in every run of the ten best configurations.

Error patterns: The gap between exact and relaxed accuracy reveals classification style (Figure 2). We identify three patterns: (1) *Decisive* models like gpt-5-nano (gap=11.6%) and grok per_publisher (78.4% exact, gap 13.7%) commit to clear labels with few yes→partial errors (27 and 67 respectively). (2) *Balanced* models like gemini-2.5-flash-lite per_publisher (gap=15.9%) show moderate hedging. (3) *Conservative* configurations such as gemini-2.5-flash-lite with the `label_only_no_rules` per_claim_publisher prompt (gap=26.5%, about 155 yes→partial per run) inflate relaxed accuracy through excessive hedging; with the rule-bearing prompts the same model hedges far less. The embedding baseline fails differently: 130 not_mentioned→partial errors from inability to distinguish semantic absence from low similarity.

5 Conclusion, Limitations and Future Work

On 1,079 annotated claim–publisher pairs, the per_publisher architecture with the reduced prompt on gemini-3-flash or gemini-2.5-flash-lite gives the best accuracy of the study (82%) at a few cents per event set; the frontier models tested do not outperform it with reasoning off or at low effort; whether a prompt states the classification rules matters more than its length, or than the temperature for the one model swept; and every LLM configuration is above the embedding baseline, the best by 25 pp.

Annotation, language and contradiction: Ground truth was created by a single annotator, so inter-annotator agreement cannot be quantified, and `partially` is subjective on both of its boundaries, with `yes` and with `not_mentioned` (36% of wrong `partially` labels fall on `not_mentioned` pairs); misclassification can falsely attribute a stance to a publisher or over-read `partially` as endorsement, so high-stakes uses need human review. The claims were extracted by gemini-3-flash (Section 3.2), itself one of the classifiers, which may favour its own phrasing; its first place should be confirmed with another extractor. Results are specific to Slovenian news. The extremely rare `no` class (0.1%, n=1) prevents meaningful evaluation of contradiction detection—the key capability distinguishing LLMs from embedding baselines; in FNC-1 the analogous *disagree* class remained the hardest even with hundreds of examples [10].

Reasoning budget and run coverage: Every model except grok-4.1-fast was evaluated with reasoning off, minimal or low; the finding that budget-tier models match frontier models therefore holds *at low deliberation*. The constraint is not neutral: models built around deliberation (the GPT-5 family, gpt-5.2 in particular) plausibly lose more from it than the others, so RQ2 is the finding most exposed to it and should be read as a statement about low-deliberation operation, not about model capability. Runs were made without seed control; the T=0 configurations and grok-4.1-fast were run only once (Section 3.6).

Three directions would strengthen these findings: multiple annotators, to measure agreement on both boundaries of `partially`; a decision on whether a contradiction (`no`) class is meaningful for news coverage at all and, if it is, events curated to contain explicit contradictions; and a sweep of reasoning effort (low/medium/high) for the top configurations, to test whether the budget-versus-frontier finding survives higher deliberation.

Acknowledgments

The research was supported by the Slovenian Research and Innovation Agency through the research program Knowledge Technologies (P2-0103) and the Gravity project AI for Science (GC-0001) and European HE projects TWON (GA 101095095) and Raido (GA 101135800). The authors acknowledge usage of LLMs in grammar refinement and overall manuscript organization and take full responsibility for the final content of this publication.

References

[1] Georgi Trajkov, Marko Grobelnik, and Adrian Mladenić Grobelnik. 2025. BetweenTheLines - cross source news analysis. In *Proceedings of the 28th International Multiconference (SiKDD)*. https://is.ijs.si/wp-content/uploads/2025/11/IS2025_Volume-C.pdf.
[2] Alphaeus Dmonte et al. 2024. Claim verification in the age of large language models: a survey. arXiv: 2408.14317 [cs.CL].
[3] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10, 178–206.
[4] Damian Trilling and Marieke van Hoof. 2020. Between article and topic: news events as level of analysis and their computational identification. *Digital Journalism*, 8, 10, 1317–1337.
[5] Federico Errica et al. 2025. What did i do wrong? quantifying LLMs' sensitivity and consistency to prompt engineering. In *Proceedings of NAACL-HLT (Volume 1: Long Papers)*, 1543–1558.
[6] Matthew Renze and Erhan Guven. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of EMNLP*, 7346–7356.
[7] Samuel R. Bowman et al. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, 632–642.
[8] Gregor Leban et al. 2014. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web* (WWW '14 Companion), 107–110.
[9] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: sentence embeddings using siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, 3982–3992.
[10] Andreas Hanselowski et al. 2018. A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of COLING*, 1859–1874.

A Hybrid Lexicon-LLM Approach to Skill Extraction from Job Advertisements

Natalija Mečkaroska
natalijamechkaroska@gmail.com
University of Ljubljana
Ljubljana, Slovenia

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

Converting the free-text requirements of job advertisements into structured skill data is a bottleneck for labour-market applications. Two approaches are available without annotated training data: matching against a controlled vocabulary, or prompting a large language model (LLM). Which is preferable has not been established on human-annotated data, since existing comparisons score against benchmarks derived from the taxonomy itself. We compare a lexicon built from the 13,939 skills of the European ESCO taxonomy against three open-weight LLMs, on a manually annotated corpus of 60 advertisements containing 744 skill annotations, scored by concept-level precision, recall and F1 under exact and relaxed matching with McNemar's test. The lexicon is the most precise system (P = .519) while every LLM recalls more (R = .43 to .52), and pairing the lexicon with the best-calibrated LLM outperforms any component alone (F1 = .463). The three LLMs differ sharply in output volume, predicting 13.2, 25.5 and 28.0 skills per advertisement against 12.4 in the reference annotations, so over-generation is a property of a particular LLM rather than of generative extraction. LLMs agree with each other far more than with the lexicon (Jaccard .66–.69 against .21–.22), so the hybrid gain comes from combining paradigms rather than LLMs. The annotated corpus and code are publicly released.

Keywords

skill extraction, ESCO, large language models, information extraction, labour market analysis

1 Introduction

Online job advertisements are among the richest signals about labour demand, but they are unstructured text. An advertisement asking for "solid experience with containerised deployments" states a skill that the European Skills, Competences, Qualifications and Occupations classification (ESCO) [6], the standard European reference vocabulary and the target taxonomy of most recent work [2, 3, 14], records under a different label. Any system reasoning over skills depends on an upstream extraction step whose quality bounds everything downstream. This study arose from one such system: a skill-based job recommender for students, which matches a candidate's skills against those an advertisement requires. It consumes sets of skills rather than spans, which is why we annotate at the level of skill concepts.

Several approaches have been applied, including supervised span labelling [13], retrieval against a taxonomy [3, 14] and generative extraction [2]. We focus on the two available without annotated training data. A lexicon matches surface forms from a controlled vocabulary: transparent, free and auditable, but able to find only what the vocabulary contains. An LLM extractor prompts a large language model (LLM) to report the skills an advertisement states: robust to unanticipated wording, but costly at scale, non-deterministic and prone to reporting implied rather than stated skills. The two fail for different reasons, so if those failures are independent, combining them should recover more than either alone. We test this directly.

Our contributions are: (i) a controlled comparison of lexicon-based, LLM-based and hybrid extraction on a common corpus with significance testing; (ii) a manually annotated corpus of 60 advertisements with 744 concept-level skill annotations, released publicly; and (iii) an analysis of where lexicon false positives originate.

2 Related work

Annotated resources for skill extraction are scarce and sit below the document. Zhang et al. [13] introduced SkillSpan, a span-level dataset without taxonomy links; Decorte et al. [3] later labelled those spans with ESCO concepts and released a sentence-level subset. Senger et al. [12] survey the field and find annotation the binding constraint throughout. Our corpus is smaller but records the ESCO concepts an entire advertisement states, the unit a recommender consumes, and we are not aware of a comparable resource.

A second line treats extraction as taxonomy linking. Decorte et al. [3] show that drawing negatives from related ESCO skills yields the largest gains in distantly supervised extraction, and Zhang et al. [14] formulate the task as entity linking. More recently the taxonomy has been approached generatively: Clavié and Soulié [2] build a zero-shot ESCO matcher without human annotations, and Decorte et al. [4] use LLMs to generate training data for a classifier. These establish that LLM-driven pipelines are strong here, but not how a plain zero-shot prompt compares against the simplest deployable alternative, since both evaluate against benchmarks derived from ESCO itself rather than independent human judgement.

Combining a curated lexicon with a statistical component is a recurring pattern where annotated data is scarce [7], and the same comparison has been made in other extraction domains. Brank et al. [1] compare an expert-curated keyword index against an LLM for detecting olfactory references in Slovenian heritage texts, using the model family we adopt, and report that the keyword method returns higher frequencies at the cost of noise while the LLM is sparser but more context-sensitive, the two agreeing on about a quarter of mentions. Their corpus carries no human annotations, so where the methods disagree neither can be shown correct. Supplying that missing reference point is what this paper contributes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.84

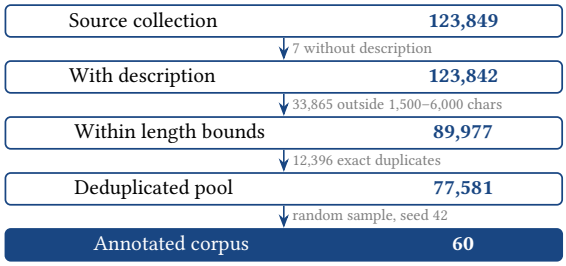


Figure 1: Construction of the evaluation corpus.

3 Data and annotation

3.1 Source corpus

We draw advertisements from a publicly available collection of 123,849 LinkedIn job postings published in 2023–2024 [8]. The collection is predominantly United States-based: 58% of the annotated advertisements contain US-specific markers. ESCO is used as a normalisation vocabulary rather than a jurisdictional standard, supplying a stable label space against which all systems are scored identically. Descriptions contain the complete advertisement text, median length 3,435 characters; we use a static collection rather than a live aggregator API because the two APIs we tested return only truncated descriptions.

3.2 Inclusion and exclusion criteria

No language filter was applied; the sampled advertisements are all English, which reflects the composition of the source collection rather than a selection decision. Four filters were applied, in order. (1) Postings with a missing description were discarded, leaving 123,842. (2) Descriptions were restricted to between 1,500 and 6,000 characters. (3) Exact-duplicate descriptions were removed, retaining the first occurrence. (4) A random sample of 60 postings was drawn from the remaining pool with a fixed seed. Of the 31 fields we retain five (identifier, title, company, location, description). Descriptions were then normalised by unescaping HTML entities and collapsing whitespace; because this ran after filtering, a few fall marginally below the lower bound (minimum 1,493 characters).

3.3 Rationale and effect on the sample

The lower length bound excludes postings that are stubs rather than complete advertisements: the collection contains descriptions as short as 2 characters, consisting of a role summary with no requirements section. The upper bound limits annotation cost, the binding constraint on corpus size, and duplicate removal prevents one posting contributing its skills twice.

Of the 123,842 postings with a description, 89,977 fall within the length bounds and 77,581 remain after deduplication, so the sample was drawn from 63% of the collection. Mean description length is 3,766 characters in the collection against 3,458 in the filtered sample, so the results characterise mid-length postings specifically.

3.4 Annotated subset

The evaluation corpus is the 60 annotated advertisements. They have a mean length of 3,355 characters (median 3,152) and contain 744 skill annotations, a mean of 12.4 per advertisement. Figure 1 summarises the construction of the corpus.

3.5 Annotation scheme

Extraction can be evaluated at the span level, marking character positions of each mention, or at the concept level, recording only the set of skills an advertisement states. We annotate at the concept level, reflecting the downstream use and reducing annotation cost.

We manually annotated each advertisement with the skills it states, following rules fixed before annotation began: a skill is recorded when the advertisement states it as a requirement, preference or responsibility; job titles are not skills; skills a reader would infer but the advertisement does not state are excluded; skills appearing only in equal-opportunity statements, benefits sections or company descriptions are excluded; and mentions with no adequate ESCO counterpart are recorded in the advertisement's own wording.

Annotation was performed by one annotator working from a sheet pre-populated with the lexicon's predictions, which the annotator corrected. This is post-editing rather than independent annotation, and its dependence on the lexicon is measurable: of the 391 proposals shown, 188 (48%) were rejected, and 541 of the 744 final annotations (73%) were added by the annotator. The set is therefore not a restatement of the lexicon's output, though neither is it independent of it; Section 6 discusses the residual bias.

4 Methods

4.1 Task definition

Each system takes an advertisement and returns a set of skills, compared against the reference annotations recorded for that advertisement. All outputs are compared as normalised strings, so every system is scored in the same way.

4.2 Lexicon-based extraction

We construct a lexicon from the ESCO skills classification, taking each concept's preferred label together with its alternative labels as surface forms and discarding forms of two characters or fewer: 13,939 concepts and 100,331 distinct surface forms. Extraction is case-insensitive n-gram matching against the advertisement text, with matched forms mapped back to their concept. The system requires no training data and is deterministic and auditable.

4.3 LLM-based extraction

We evaluate three open-weight instruction-tuned LLMs: Llama 3.3 70B Instruct [10], DeepSeek V3 [5] and Qwen 2.5 72B Instruct [11]. All are open-weight and available through hosted APIs, representative of what a practitioner can deploy without licensing a proprietary LLM; Llama 3.3 70B was chosen additionally for its use in comparable keyword-versus-LLM work [1]. All three were served through the OpenRouter API and prompted identically, zero-shot at temperature 0, requesting a JSON list of skill names, so differences reflect the LLMs rather than the serving configuration. The prompt instructs each LLM to list only skills the advertisement states, exclude the job title, and ignore skills appearing only in equal-opportunity statements, benefits sections or legal disclaimers, mirroring the annotation guideline. No advertisement approaches the LLMs' input limits. Responses failing to parse are retried once. Figure 2 gives the prompt verbatim.

```
You are extracting skills from a job advertisement.
List every skill, tool, technology, programming language, certification,
or professional competence that the advertisement states as a requirement,
preference, or responsibility of the role.
Rules:
- Only list skills the advertisement actually states. Do not infer skills.
- Do not list the job title itself.
- Ignore skills mentioned only in equal-opportunity statements, benefits
sections, or legal disclaimers.
- Use short skill names (1–4 words), lowercase.
Return ONLY a JSON array of strings. No explanation, no markdown.
```

Figure 2: The zero-shot extraction prompt.

	Relaxed			Exact		
System	P	R	F1	P	R	F1
Lexicon (ESCO)	**.519**	.273	.358	**.512**	.269	.352
Llama 3.3 70B	.400	.426	.412	.256	.273	.264
DeepSeek V3	.251	**.516**	.338	.163	.335	.219
Qwen 2.5 72B	.208	.469	.288	.109	.247	.152
Union (lex+Llama)	.384	.583	**.463**	.310	.472	**.374**
Union (lex+DeepSeek)	.261	.653	.373	.211	**.528**	.301
Union (lex+Qwen)	.226	.621	.332	.173	.476	.254

Table 1: Concept-level P, R and F1 on 60 advertisements (744 reference annotations).

4.4 Normalisation and hybrid combination

Because the LLMs emit free-form strings, all outputs, reference annotations included, pass through the same normalisation (lowercasing, punctuation stripping, whitespace collapsing), so no system is advantaged by the wording in which it expresses a skill. We additionally evaluate the union of the lexicon with the best-calibrated LLM, which operationalises the complementarity hypothesis of Section 1: if the two paradigms fail on different skills, their union should recover more than either component.

5 Evaluation and results

5.1 Metrics

We report micro-averaged precision, recall and F1, pooling true positives, false positives and false negatives across advertisements. Because systems express the same skill in different surface forms, we report two matching regimes. Exact matching requires string identity after normalisation. Relaxed matching additionally accepts a prediction when one string contains the other, or when token-level Jaccard similarity with a reference annotation is at least 0.6; for example, the output skill "read drawings" matches the reference annotation "read technical drawings" (Jaccard 2/3). Each reference annotation is matched at most once. Relaxed matching cannot prevent a loose match crediting a prediction a human would judge incorrect, so relaxed figures should be read as an upper bound and exact figures as a lower bound. Significance is assessed with McNemar's test [9], in its chi-square form with continuity correction, over the 744 (advertisement, reference annotation) instances.

5.2 Main results

The two paradigms separate cleanly. Table 1 reports both matching regimes. The lexicon is the most precise system under both; every LLM recalls more once wording differences are tolerated, DeepSeek most (.516 against .273). The union of the lexicon with Llama is the strongest configuration (F1 = .463 relaxed, .374 exact), recovering 58% of the reference annotations against the lexicon's 27%.

Comparison	b	c	χ^2	p
Lexicon vs Llama	113	227	37.6	$< 10^{-9}$
Lexicon vs DeepSeek	99	280	85.5	$< 10^{-15}$
Llama vs DeepSeek	31	98	33.8	$< 10^{-8}$
Llama vs Qwen	53	85	7.0	.008
DeepSeek vs Qwen	86	51	8.4	.004

Table 2: McNemar tests under relaxed matching, over 744 paired instances. b and c are instances retrieved by only the first and only the second system.

The three LLMs differ sharply in how much they output, though not in what they find. The lexicon predicts 6.5 skills per advertisement, Llama 13.2, DeepSeek 25.5 and Qwen 28.0, against 12.4 in the reference annotations. Llama is calibrated close to the annotation density while the other two over-generate and lose precision accordingly, so over-generation is a property of a particular LLM rather than of generative extraction as such. The surplus is systematic: the other two most often add dispositions (*integrity*, *positive attitude*, *adaptability*) and employment conditions that are not competences (*travel*, *lifting*, *background check*). Qwen's phrasing is also longer, 35% of its predictions running to three or more words against 22% of the reference annotations. Pairing the lexicon with either over-generating LLM raises recall (.653 and .621) but costs more precision than it gains.

A second difference concerns vocabulary rather than volume. The lexicon emits ESCO labels by construction, but the LLMs were never told to target the taxonomy: only 12.8% of Llama's predictions, 9.3% of DeepSeek's and 5.6% of Qwen's match an ESCO surface form, an ordering that matches the F1 ranking exactly. Predictions that do match ESCO are roughly twice as likely to be correct (65.7% against 36.1% for Llama, 55.2% against 21.9% for DeepSeek, 53.2% against 18.7% for Qwen). The reference annotations are themselves only 39% ESCO-recognised, so full coverage is not the target; the LLMs fall well short even of that. On 3 of the 60 advertisements the lexicon retrieved nothing.

5.3 Significance

The matching regime matters more for the LLMs than for the lexicon: relaxing the match adds 114, 135 and 165 true positives respectively, against 3 for the lexicon, because the lexicon emits canonical ESCO labels by construction while the LLMs paraphrase. Scoring by string identity therefore understates generative systems.

All five pairwise differences are significant, including those between the three LLMs (Table 2). We control the family-wise error rate with the Holm–Bonferroni procedure; all five remain significant at $\alpha = .05$, the weakest being Llama vs Qwen (adjusted $p = .008$). The union's recall gains are descriptive rather than separate hypothesis tests: under the same one-to-one matching it retrieves 231 more annotations than the lexicon and 117 more than Llama, while losing none, as follows by construction from taking the union. Complementarity is also visible in agreement between the sets of reference annotations retrieved by each system across the corpus: the three LLMs have pairwise Jaccard similarity .66 to .69, compared with .21 to .22 between each LLM and the lexicon. The LLMs retrieve similar annotations, whereas the lexicon retrieves different ones, which is why pairing paradigms helps more than pairing two LLMs.

5.4 Sources of lexicon false positives

Lexicon false positives are concentrated: *plan* (13), *manage time* (12), *think creatively*, *morality* and *history* (9 each) account for 52

of 188 false-positive instances (28%). Inspection shows most are triggered outside the requirement sections: "pregnancy", "morality" and "genetics" match equal-opportunity statements; "history" matches company descriptions and the phrase "work history"; "energy" and "electronics" match benefits copy. In one case an ESCO concept was matched against a query parameter inside an application URL.

5.5 Runtime and cost

Extraction required 4.6 s per advertisement for Llama, 7.2 s for DeepSeek and 7.8 s for Qwen; output ranged from 3,534 tokens for Llama to 10,068 for Qwen, mirroring the over-generation above. All three produced valid JSON on every advertisement, with zero parse failures across 180 extractions. The three runs cost approximately $0.01, $0.03 and $0.02 for Llama, DeepSeek and Qwen respectively, about $0.06 in total for the corpus, while the lexicon runs locally in milliseconds at no cost.

6 Discussion

Neither paradigm is adequate alone. The best single system reaches F1 = .412 and the hybrid .463, which is low in absolute terms and reflects both the difficulty of the task and the deliberately simple configurations we test.

The lexicon's weakness is contextual, not lexical: its errors reflect blindness to where a phrase occurs and in what sense, rather than missing vocabulary. "Manage time" is a genuine requirement in a retail lead role and a false positive in four advertisements where no such requirement appears; "metrology" is correct for a role selling measurement instrumentation and wrong for a payments executive. Restricting matching to requirement-bearing sections, or excluding a stop-list of over-general concepts, would remove many false positives without changing the method.

The LLMs' weakness is calibration to the annotation scheme, and it varies sharply between them. Llama produces approximately the right number of skills per advertisement. DeepSeek and Qwen produce twice as many, splitting single requirements into several skills and reporting attributes from benefits sections. Part of the disagreement is an artefact of our design: the prompt never mentions ESCO, so the LLMs are scored against a vocabulary they were not told to target, and only 5–13% of their output reaches it.

Because the two paradigms fail differently, their union recovers more than either. Where missing a required skill is costlier than carrying a spurious one, as in job recommendation, a hybrid extractor is preferable to either component alone.

6.1 Limitations

The generative side was evaluated under one zero-shot configuration: the prompt did not mention ESCO, give examples, or constrain output to ESCO candidates. Because so little LLM output is expressed as ESCO terms, exact matching has limited interpretive value for them; the large gain under relaxed matching (114, 135 and 165 added true positives against 3 for the lexicon) confirms this. The results therefore characterise these LLMs under one deliberately simple prompt rather than generative extraction in general. The hybrid union's superiority is likewise not guaranteed under stronger configurations that address these limitations; better calibration or ESCO alignment could change the balance between the components.

The corpus is small (60 advertisements), English-language, and all systems were evaluated only on advertisements passing the filters of Section 3.2. Short postings tend to lack the requirements sections all systems depend on, so the lexicon's recall on them is plausibly lower than reported.

Annotation was performed by a single annotator, so no agreement statistic is available, and the reference set was produced by post-editing the lexicon's predictions, so it is not independent of one of the systems it scores. The bias runs in a known direction: a retained false positive inflates the lexicon's precision, and a skill it never proposed and the annotator did not add is absent from the reference set, inflating its recall. The reported lexicon precision of .519 should be read as an upper bound. The same bias understates every LLM, so the finding that all three recall more than the lexicon is conservative, as is the hybrid's advantage over the lexicon; its advantage over Llama is not, because the union contains the lexicon's output and so shares the bias in its favour. Annotating a further subset with no system output visible would quantify this directly.

7 Conclusion and future work

We compared lexicon-based and LLM-based skill extraction on a manually annotated corpus of 60 job advertisements with 744 skill annotations. The lexicon is most precise (.519), all three LLMs recall more (.43 to .52), and pairing the lexicon with the best-calibrated LLM is the strongest configuration (F1 = .463). Five ESCO concepts account for 28% of lexicon false positives, most triggered by text outside the requirement sections.

Three directions follow. Restricting lexicon lookup to passages containing requirements is a minimal change that would remove many false positives. The generative side was left in its simplest form: naming ESCO in the prompt, adding few-shot examples, or re-ranking output against retrieved ESCO candidates are untried, and the gap between ESCO-recognised and unrecognised predictions suggests a validation layer alone would recover much of the lost precision. Finally, the extracted skills feed the graph-based candidate–vacancy matching of the recommender this work was built for.

The annotated corpus, the extraction and evaluation code, and the full result files are available at https://github.com/naatteeee/skill-extraction.

Acknowledgements

The research was supported through the European project Raido (GA 101135800). The authors acknowledge usage of LLMs in grammar refinement and overall manuscript organization and take full responsibility for the final content of this publication.

References

[1] Janez Brank, Inna Novalija, Dunja Mladenić, and Marko Grobelnik. 2025. LLM based approach to extracting smells in Slovenian corpora. In *Proceedings of Data Mining and Data Warehouses (SiKDD 2025)*. DOI: 10.70314/is.2025.sikdd.5.

[2] Benjamin Clavié and Guillaume Soulié. 2023. Large language models as batteries-included zero-shot ESCO skills matchers. In *Proceedings of the 3rd Workshop on Recommender Systems for Human Resources (RecSys-in-HR 2023)* (CEUR Workshop Proceedings). Vol. 3490. arXiv: 2307.03539.

[3] Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Design of negative sampling strategies for distantly supervised skill extraction. In *Proceedings of the 2nd Workshop on Recommender Systems for Human Resources (RecSys-in-HR 2022)* (CEUR Workshop Proceedings). Vol. 3218.

[4] Jens-Joris Decorte, Severine Verlinden, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Extreme multi-label skill extraction training using large language models. In *International Workshop on AI for Human Resources and Public Employment Services (AI4HR&PES) at ECML-PKDD 2023*. arXiv: 2307.10778.

[5] DeepSeek-AI. 2024. DeepSeek-V3 model card. Hugging Face. (2024).

[6] European Commission. 2025. ESCO v1.2.1. esco.ec.europa.eu. (2025).

[7] Sofija Kochovska, Branko Kavšek, and Jernej Vičič. 2025. A hybrid lexicon-machine learning approach to Macedonian sentiment analysis. In *Proceedings of Data Mining and Data Warehouses (SiKDD 2025)*.

[8] Arsh Koneru. 2024. LinkedIn job postings (2023–2024). Kaggle. CC BY-SA 4.0. (2024).

[9] Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12, 153–157. DOI: 10.1007/BF02295996.

[10] Meta AI. 2024. Llama-3.3-70B-Instruct model card. Hugging Face. (2024).

[11] Qwen Team. 2024. Qwen2.5-72B-Instruct model card. Hugging Face. (2024).

[12] Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. 2024. Deep learning-based computational job market analysis: a survey on skill extraction and classification from job postings. In *Proceedings of NLP4HR 2024*, 1–15. DOI: 10.18653/v1/2024.nlp4hr-1.1.

[13] Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank. 2022. SkillSpan: hard and soft skill extraction from English job postings. In *Proceedings of NAACL-HLT 2022*, 4962–4984. DOI: 10.18653/v1/2022.naacl-main.366.

[14] Mike Zhang, Rob van der Goot, and Barbara Plank. 2024. Entity linking in the job market domain. In *Findings of EACL 2024*, 410–419. DOI: 10.18653/v1/2024.findings-eacl.28.

Benchmarking Machine-Learning Forecasts of Low-Magnitude Seismicity in the Italy–Slovenia Border Region

Oleksandra Topal
oleksandra.topal@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Erik Calcina
erik.calcina@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Luciano Telesca
luciano.telesca@cnr.it
Institute of Integrated Methodologies for Earth Observation, CNR
Tito Scalo, Italy

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Reliable earthquake forecasting is an important scientific challenge with direct relevance to seismic monitoring and preparedness. Neural and tree-based models were benchmarked on 5784 earthquakes from the Italy–Slovenia cross-border seismic area (magnitudes 1.15–4.04, 2016–2023), using catalogue variables, satellite data and weather reanalysis. Three experiments examined next-earthquake regression for magnitude, epicentre location and inter-event time; earthquake-occurrence classification with an evaluation audit; and the same classification task in an expanded experiment with statistical assessment. Regression remained near simple-baseline performance: epicentre prediction improved by only 1.6% beyond recent-location averaging, with no further gain from the additional inputs. The evaluation audit showed that the strongest classification score was supported by repeated forecasts associated with one earthquake. The expanded experiment identified a possible five-day signal at magnitude 2.5 and above with support from 22 distinct earthquakes, but the result remained uncertain after all model configurations were considered, and the predicted probabilities did not match observed frequencies. Credible forecasting claims therefore require appropriate baselines, explicit counts of distinct earthquakes, dependence-aware statistical testing and calibration.

Keywords

earthquake forecasting, machine learning, low-magnitude seismicity, regression, classification, forecast evaluation

1 Introduction

The continuous movement of tectonic plates deforms the Earth's crust and accumulates stress that is intermittently released by earthquakes, most of which are low in magnitude. This study investigates whether low-magnitude seismicity can support reliable machine-learning forecasts in the Italy–Slovenia cross-border seismic area and its immediate surroundings (Figure 1). Earthquakes were selected from the Tarchini et al. catalogue [9] within a polygon covering the principal concentration of seismicity across the Italy–Slovenia border and at moment magnitude $M_w \geq 1.15$, the more conservative of two completeness estimates from a Gutenberg–Richter analysis [8]. The resulting study catalogue contains 5784 earthquakes, only 42 of which have $M_w \geq 3$, limiting the support available for evaluating higher-magnitude forecasts.

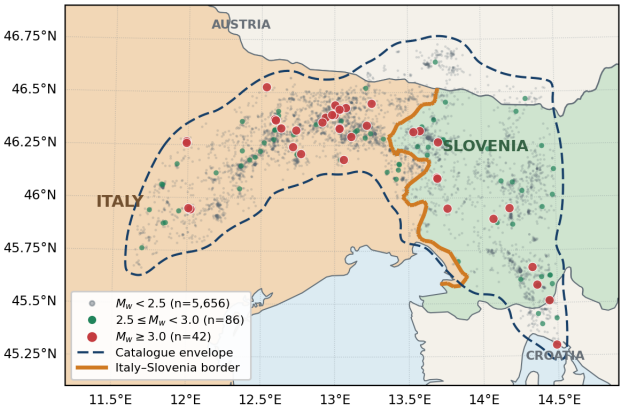


Figure 1: Spatial distribution of the 5784 earthquakes in the Italy–Slovenia cross-border seismic area and its immediate surroundings. The dashed outline marks the extent of the selected catalogue; national boundaries are shown for geographic orientation (Natural Earth 1:10m).

Two tasks were examined: next-earthquake regression for magnitude, epicentre location and inter-event time, and classification of whether an earthquake at or above a specified magnitude would occur within a fixed forecast horizon. Classification was evaluated through an audit of distinct supporting earthquakes and an expanded statistical assessment. Benchmarking against simple catalogue-based predictors extends event-level work on Italian seismicity [5] and responds to evidence that model complexity does not necessarily provide greater predictive skill [7].

2 Methodology

2.1 Regression of Next-Earthquake Properties

An earthquake event denotes one catalogue record with an origin time, epicentre location, depth and magnitude; *event* is used below as its shorter form. The regression experiments followed a staged design in which progressively richer representations were evaluated against simple catalogue baselines (Figure 2). In the initial stage, the C7 representation (Table 1) combined magnitude, latitude, longitude, depth and inter-event time with two cyclical variables encoding the earthquake origin time on a 24-hour cycle, and was evaluated using convolutional neural network–long short-term memory (CNN–LSTM) and random-forest models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.98

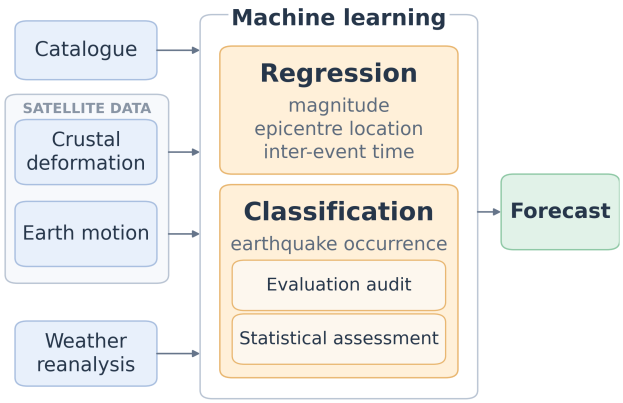


Figure 2: Benchmarking pipeline for next-earthquake regression and earthquake-occurrence classification using catalogue, satellite and weather-reanalysis data.

Because the catalogue-only models provided no improvement beyond the baselines, C7 was extended to M50 and evaluated with the same learners. M50 augmented C7 with a rolling Gutenberg–Richter b-value, which describes the relative frequency of small and larger earthquakes, its short-to-long-term change, and simplified estimates of tidal phase, amplitude and rate of change, together with 38 variables from additional data: 20 daily Global Navigation Satellite System (GNSS) displacement variables from ten stations [3], 11 European Ground Motion Service (EGMS) Interferometric Synthetic Aperture Radar (InSAR) motion variables for 2018–2022 [4], and 7 daily variables from the fifth-generation European Centre for Medium-Range Weather Forecasts reanalysis (ERA5) [6]. Same-day ERA5 matching, full-period EGMS summaries and full-catalogue median imputation made this a retrospective experiment.

Whereas C7 and M50 preserved each sliding event window as a sequence, the final stage transformed every M50 window into the tabular S300 representation by computing six summary statistics for each variable, thereby enabling evaluation with the gradient-boosted tree models CatBoost, LightGBM and XGBoost. Because the external sources were introduced together and S300 was evaluated with different learners, the experiment compared complete configurations rather than isolating the contribution of GNSS, EGMS or ERA5.

Table 1: Feature representations used for next-earthquake regression.

Set	Size	Representation
C7	7	Magnitude, latitude, longitude, depth, inter-event time and cyclical origin hour (2 variables)
M50	50	C7; b-value/change; tidal phase/amplitude/rate; 20 GNSS; 7 ERA5; 11 EGMS InSAR variables
S300	300	Six history summaries for each M50 variable

Regression samples used a sliding event window of the N preceding earthquakes to predict the next, advancing one earthquake at a time; models were trained through 2021 and evaluated in 2022–2023, which also guided model and input-history selection and S300 early stopping rather than serving as an untouched holdout. C7 and M50 each tested 11 values of N spanning 50–1500 and 50–2500 events, respectively, whereas S300 tested only $\{500, 1250, 2500\}$, leaving 500 and 1250 as the shared settings. Performance was measured by root-mean-square error (RMSE) against two baselines: the training mean, which predicts the mean of the target over the training period, and a moving average over the preceding N target values, representing recent-history persistence. The moving average was computed for C7 and M50, and its best RMSE is reported in Table 3 as a common reference. Epicentre RMSE used a Euclidean latitude–longitude norm in degrees; C7 and M50 values were multiplied by $\sqrt{2}$ to match the S300 convention.

2.2 Earthquake-Occurrence Classification: Evaluation Audit

The limited improvement of next-earthquake regression over simple baselines motivated a coarser forecasting question: whether at least one earthquake above a selected magnitude would occur within a fixed forecast horizon. The first classification experiment also used sliding event windows, issuing one prediction after each catalogue event. Each window used the preceding N events as its input history and the following D days as its *forecast horizon*. For an event at time t_i, the binary label is

$$y_i(M, D) = \mathbf{1}\{\exists j > i : t_j \le t_i + D,\ M_{w,j} \ge M\}, \qquad (1)$$

where M is the magnitude threshold. A window is positive when at least one earthquake meeting that threshold occurs within its forecast horizon; such a future earthquake is called a *target earthquake*, and the window is negative when no such earthquake occurs. Performance was summarised by the area under the receiver-operating-characteristic curve (AUC). Because consecutive windows share $N - 1$ input events and their forecast horizons overlap, one target earthquake may produce several consecutive positive windows; the audit therefore linked each positive window to its target and counted the distinct target earthquakes.

The audit evaluated an artificial neural network (ANN), support vector machine (SVM) and random forest using input histories of $N = 100$, 200 or 300 events and forecast horizons of $D = 10$, 30, 60, 90 or 120 days. The whole-catalogue experiment used the full study catalogue with 99 variables summarising past seismicity: b-value and inter-event-time statistics, a magnitude histogram, and Allan-factor and detrended-fluctuation measures. It tested $M \in \{3.0, 3.5\}$, giving 90 model combinations.

To reduce the contribution of clustered seismic sequences, the audit was repeated with a declustered catalogue containing 1717 earthquakes at $M_w \ge 1.5$, obtained by nearest-neighbour declustering [10]. The declustering was not re-run here, so the catalogue was used as a fixed data product. Because only 11 of these earthquakes reach $M_w \ge 3.5$, the threshold $M = 2.5$ was added for measurable target support, giving 135 combinations with a related 95-variable representation. Each combination was repeated 100 times on a fixed chronological 70/30 split, with seeds affecting only model initialisation and class balancing.

2.3 Earthquake-Occurrence Classification: Statistical Assessment

The expanded assessment addressed the same occurrence question on the declustered catalogue, broadening the model comparison and adding First-In-Cluster (FIC) evaluation. FIC retains every negative window but only the first window in each consecutive run of positive windows; here *cluster* means adjacent positive labels, not a seismic cluster. Its AUC reduces the repeated weighting of positive forecasts without establishing independence, because the retained negative windows can still overlap.

Performance was reported as all-window AUC, FIC AUC and the number of distinct target earthquakes, against a causal persistence baseline that forecasts occurrence in the next D days whenever a qualifying earthquake occurred in the previous D days. Model scores were held fixed while labels were permuted

independently, in contiguous 20-row blocks, or circularly shifted, using up to 100 000 permutations (Table 2); Benjamini–Hochberg correction [2] gave the reported q-values. Calibration in the large was assessed by comparing the sum of predicted probabilities with the observed number of positive windows.

Table 2: Design of the expanded statistical assessment.

Factor	Values or procedure
Threshold M	2.5, 3.0, 3.5
Forecast horizon D	5, 7, 10, 15, 50 days
Input history N	50, 100, 200, 300, 500 events
Features	Seis26; Asim60 [1]; Full86
Learners	Nine linear, kernel, tree-based and neural models
Development	Three walk-forward folds, 2016–2021; decision-threshold choice and prioritisation
Evaluation	Full 2025-configuration grid scored on 2022–2023
Inference	Causal baseline; independent, 20-row block and circular-shift label permutations; bootstrap FIC AUC interval; within-family and global Benjamini–Hochberg correction

Seis26 denotes 26 timing, distance, Gutenberg–Richter, magnitude and moment features; Asim60 denotes 60 energy, timing, seismicity, magnitude-deficit, residual and recurrence features [1]; and Full86 concatenates both sets. The nine learners included logistic regression, elastic net, SVM, k-nearest neighbours (KNN), random forest, Extra Trees, XGBoost, histogram-based gradient boosting (HistGBM) and a multilayer perceptron. The $5 \times 3 \times 5 \times 3 \times 9 = 2025$ grid gave 135 model combinations per threshold–horizon family, and the comparison is exploratory because it was scored on the period used for selection.

3 Results

3.1 Regression of Next-Earthquake Properties

Table 3 reports the forecast error for each target against the simple baselines. Neither magnitude nor inter-event time separated from them; the largest inter-event-time gain was under a minute (13.832 versus 13.844 h). Only epicentre improved, and most of that was persistence: the 4.4% gain over the training mean fell to 1.6% against the moving average, and the near-identical C7 and M50 values showed that GNSS, EGMS and ERA5 added no location information. The best epicentre error was roughly 58–84 km at this latitude.

Each column reports the lowest RMSE attained by that representation over its own sweep of N. C7 and M50 searched 11 input-history lengths, compared with three for S300; learners also differed, with C7 and M50 using CNN–LSTM apart from random-forest epicentre estimates and S300 using LightGBM or CatBoost. The training-mean baseline also shifted with N; its smallest value was used as the strictest reference, and only for epicentre did this matter (0.793°–0.817°).

No input-history length changed this: the shortest and intermediate histories performed worse than the baselines for both magnitude and inter-event time, and only the longest matched them, reproducing the training mean rather than improving on it. Similar outcomes across sequence and tabular representations, external inputs and learners therefore indicate that the weak exact-property performance was not confined to one architecture, motivating the coarser occurrence analysis.

Table 3: Next-earthquake RMSE in the evaluation period; lower is better.

Target	Mean	Moving avg.	C7	M50	S300
Magnitude (M_w)	0.349	0.349	0.349	0.349	0.348
Inter-event time (h)	14.003	13.844	14.005	14.004	13.832
Epicentre (°)	0.793	0.770	0.770	0.769	0.758

3.2 Earthquake-Occurrence Classification: Evaluation Audit

Occurrence classification discriminated little better than chance on either catalogue: mean all-window AUC was 0.513 for the whole catalogue and 0.528 for the declustered one, with 44 of 90 and 57 of 135 model combinations below 0.5.

Table 4 relates these scores to the earthquakes behind them. Each window range spans the five forecast horizons, since a longer horizon makes more windows positive while the input-history length does not affect the label. The original results recorded positive-window counts but not the earthquakes producing them, so the labels were rebuilt from the catalogue; the rebuilt window counts matched the stored ones for all 45 combinations of input-history length, threshold and forecast horizon. Mean AUC was highest at $M = 3.5$, yet the positive windows on that row came from a single target earthquake, and the whole catalogue showed the same pattern, falling from 12 target earthquakes at $M = 3.0$ to two at $M = 3.5$. Apparent discrimination therefore coincided with the smallest event support.

Table 4: Declustered classification: mean AUC is highest at the threshold with the fewest supporting earthquakes.

Threshold M	Positive windows	Distinct earthquakes	Mean AUC
2.5	149–476	29	0.449
3.0	36–333	6	0.447
3.5	9–77	**1**	**0.688**

Individual combinations vary far more than the means provided in Table 4. At $M = 3.5$ the strongest combination was SVM with $N = 100$ and $D = 30$ d, averaging AUC 0.995 over 100 runs. Its 27 positive windows were all linked to the one target earthquake at that threshold, so the score reflects repeated forecasts before a single event rather than 27 independent successes.

3.3 Earthquake-Occurrence Classification: Statistical Assessment

The expanded assessment identified one low-threshold candidate for detailed evaluation, whose significance depended on the assumed dependence structure and whose predicted probabilities did not match observed frequencies. This candidate, HistGBM with $N = 500$, $M = 2.5$, $D = 5$ d and Full86, was selected from the 2025 model combinations (Table 5), with an all-window AUC of 0.650 and a FIC AUC of 0.707 (bootstrap 95% interval 0.580–0.823) against 0.456 for the causal persistence baseline. FIC reduced 74 positive windows among 494 to 22, which correspond to 22 distinct target earthquakes, so the all-window set carried about 3.4 positive windows per target earthquake. The baseline sat below chance not through an inverted sign but because FIC keeps the positive windows furthest from a trailing event.

Independent label permutation yielded $p = 4.6 \times 10^{-4}$ and within-family $q = 0.062$, the block null yielded $q = 0.016$, and the circular-shift null yielded $q = 0.305$; all three are reported because only the block null cleared 0.05 and it was selected from among them. Under global correction no configuration was significant, the smallest q being 0.243.

The candidate was also miscalibrated in the large: summed over the 494 windows, its predicted probabilities implied 131.1 positives against 74 observed, consistent with a base-rate shift from 0.191 in training to 0.150 in the evaluation period.

Its support is likewise bounded by the search design, since q-values correct within a threshold–horizon family rather than across the complete grid and the configuration was selected on

the period it was scored on. The higher thresholds add no support of their own, resting on six and one retained positive windows with no qualifying cell in 14 of the 15 families. Nearby input histories of $N = 300–500$ produced similar FIC AUCs without reaching $q < 0.05$ under any null, and the candidate therefore requires evaluation in a later chronological period not used for model selection.

Table 5: Selected results from the statistical assessment. The first row is the candidate; the others illustrate high AUC with sparse support. n_{FIC} is the number of retained positive windows. q is within-family under the independent label permutation; the block and circular-shift nulls give 0.016 and 0.305 for the candidate.

Configuration	Model	All-window AUC	FIC AUC	q	n_{FIC}
$M = 2.5$, $D = 5$ d, $N = 500$, Full86	HistGBM	0.650	0.707	0.062	22
$M = 3.0$, $D = 15$ d, $N = 300$, Asim60	KNN	0.633	0.722	0.895	6
$M = 3.5$, $D = 5$ d, $N = 100$, Full86	Extra Trees	1.000	1.000	0.108	1

4 Discussion and Conclusion

Taken together, the experiments indicate that forecasting conclusions depended less on model complexity than on the specificity of the target and the unit of evaluation. Next-earthquake regression recovered little information beyond simple catalogue baselines, whereas the less specific occurrence task produced a possible five-day signal at magnitude 2.5 and above across 22 distinct target earthquakes; at the same time, the evaluation audit showed that the strongest high-threshold AUC was supported by 27 positive windows linked to a single earthquake. Occurrence classification could therefore appear more successful than exact-property regression, but its evidential strength depended on how many distinct earthquakes supported that discrimination.

The limited regression gains should be interpreted within the tested data integration design. Daily GNSS and ERA5 values and approximately six-day InSAR observations were aligned with event-indexed histories, which may have obscured short-lived or delayed relationships; consequently, the absence of improvement from these inputs concerns their present representation rather than their broader physical relevance. Nevertheless, recent-location averaging explained most of the apparent epicentre gain, while the inter-event-time improvement remained below one minute, indicating that the tested models learned little beyond catalogue persistence.

For occurrence classification, AUC over all windows could count the same target earthquake repeatedly because consecutive windows covered overlapping forecast horizons. FIC AUC reduced this repetition by retaining only the first positive window in each consecutive run, but it did not establish independence because negative windows could still overlap. The five-day candidate at $M = 2.5$ reached FIC AUC 0.707, yet its significance varied with the permutation scheme, no configuration survived correction across the full grid, and its predicted probabilities did not match the observed frequencies; it should therefore be regarded as exploratory rather than as evidence of operational forecast skill.

These conclusions are also bounded by evaluation-period reuse, evaluation of the complete classification grid, and independent label permutations that did not preserve temporal dependence. Moreover, the two catalogues do not begin from the same earthquakes, the declustered one starting at $M_w \geq 1.5$ and the whole one at $M_w \geq 1.15$, and the experiments on them used different thresholds and feature representations; their results are therefore reported separately rather than compared. In the study catalogue only 42 earthquakes reached $M_w \geq 3$ and none exceeded 4.04. A credible confirmation therefore requires a later chronological evaluation period not used for model selection, inputs restricted to information available when each forecast is issued, probabilities that match observed frequencies and explicit counts of distinct target earthquakes, so that predictive discrimination can be distinguished from catalogue persistence and repeated evidence around a small number of earthquakes.

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency (ARIS) via the Research Program Knowledge Technologies (grant P2-0103) and the Gravity project AI for Science (grant GC-0001), and by the bilateral CNR–JSI project 2026–2027 “Deep Learning-based Probabilistic Real-time Earthquake Prediction for the Italy–Slovenia cross-border seismic area”.

Data licences. EGMS and ERA5 contain modified Copernicus information: EGMS was cropped and downsampled to the study area, while ERA5 was spatially averaged and aggregated daily. Neither the European Union, the European Commission nor the European Centre for Medium-Range Weather Forecasts endorses these modifications or is responsible for their use.

The authors acknowledge the use of large language models for language optimisation. While these tools contributed to refining the presentation, all conceptual frameworks, analyses and interpretations remain the sole responsibility of the authors.

References

[1] Khawaja M. Asim, Sayed S. R. Moustafa, Iftikhar Azim Niaz, Eslam A. Elawadi, Talat Iqbal, and Francisco Martínez-Álvarez. 2020. Seismicity analysis and machine learning models for short-term low magnitude seismic activity predictions in Cyprus. *Soil Dynamics and Earthquake Engineering*, 130, 105932. doi:10.1016/j.soildyn.2019.105932.

[2] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57, 1, 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

[3] Geoffrey Blewitt, William C. Hammond, and Corné Kreemer. 2018. Harnessing the GPS data explosion for interdisciplinary science. *Eos*, 99. doi:10.1029/2018EO104623.

[4] European Environment Agency. 2023. European ground motion service (EGMS), level-3 ortho. Copernicus Land Monitoring Service. (2023). https://egms.land.copernicus.eu/.

[5] Jesús Gonzalez, Wen Yu, and Luciano Telesca. 2022. Gated recurrent units based recurrent neural network for forecasting the characteristics of the next earthquake. *Cybernetics and Systems*, 53, 2, 209–222. doi:10.1080/01969722.2021.1981637.

[6] H. Hersbach et al. 2023. ERA5 hourly data on single levels from 1940 to present. (2023). doi:10.24381/cds.adbb2d47.

[7] Arnaud Mignan and Marco Broccardo. 2020. Neural network applications in earthquake prediction (1994–2019): meta-analytic and statistical insights on their limitations. *Seismological Research Letters*, 91, 4, 2330–2342. doi:10.1785/0220200021.

[8] Arnaud Mignan and Jochen Woessner. 2012. Estimating the magnitude of completeness in earthquake catalogs. *Community Online Resource for Statistical Seismicity Analysis*. doi:10.5078/corssa-00180805.

[9] Gabriele Tarchini, Luca Moratto, and Angela Saraò. 2025. A comprehensive moment magnitude catalog for the northeastern Italy region. *Seismological Research Letters*, 96, 4, 2714–2723. doi:10.1785/0220240303.

[10] Ilya Zaliapin, Andrei Gabrielov, Vladimir Keilis-Borok, and Henry Wong. 2008. Clustering analysis of seismicity and aftershock identification. *Physical Review Letters*, 101, 1, 018501. doi:10.1103/PhysRevLett.101.018501.

The Effect of Label Only vs. Label Plus Description: Comparing GLiNER1 and GLiNER2 for Clinical Entity Recognition Across Five Languages

Abdul Sittar*
abdul.sittar@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Erik Novak
erik.novak@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Erik Calcina
erik.calcina@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

We study whether zero-shot multilingual clinical Named Entity Recognition (NER) benefits from a longer, descriptive entity label rather than a short name. Using the five-language European Clinical Case Corpus (E3C: English, French, Italian, Spanish, Basque), we fine-tune two model families — GLiNER1 (small) and GLiNER2 (base) — under two label conditions, a short name ("clinical entity") and the same name paired with an explanatory description, and evaluate with both exact-match and IoU-based overlap (Intersection over Union) F1 to separate boundary errors from missed entities entirely. To go beyond aggregate scores, we apply SHAP with a paired ablation check to a span-detection scoring function and present a case study in which a configuration with a higher aggregate F1 nonetheless misses a clinically salient entity that a lower-scoring configuration detects correctly. These results suggest that model-family choice, evaluation granularity, and example-level behavior each carry information a single F1 number cannot, and that longer label descriptions are not a safe default for zero-shot clinical NER.

Keywords

clinical named entity recognition, zero-shot NER, GLiNER, E3C corpus, explainability

1 Introduction

Zero-shot and few-shot NER models such as GLiNER [10] recognize arbitrary entity types by matching free-text label descriptions against spans in the input, rather than relying on a fixed label set learned at training time. This is attractive for clinical NLP in particular, where the entities of interest (symptoms, findings, diagnoses) vary by use case, by institution, and by language, and where collecting enough annotated data to train a conventional fixed-label classifier for every new setting is rarely feasible. One line of prior work addresses this scarcity directly by generating synthetic non-English clinical text [3, 7]; we instead ask whether better use of the label interface itself — rather than more training data — can close some of that gap. In practice, this raises a simple but understudied question: does giving the model a longer, more explicit description of what counts as the entity actually improve extraction, or is a short label name sufficient?

We study this question in a clinical setting — extracting a single generic "clinical entity" span type (symptoms, diagnoses, findings) — across five languages of the multilingual E3C corpus, comparing two model families (GLiNER1 and the newer GLiNER2) under two label conditions (short name vs. name+description). We report both strict exact-match F1 and a more forgiving overlap-based F1, since in a clinical review workflow a span that is off by one token at a boundary is a very different kind of error from a span that is missed entirely, and an exact-match-only evaluation conflates the two. We further use SHAP to explain what individual predictions are driven by, at the level of single words, giving a qualitative complement to the aggregate numbers and a way to sanity-check whether an aggregate F1 improvement actually reflects more robust, generalizable behaviour or is instead concentrated in a handful of easy examples.

This paper contributes (1) a controlled comparison of short-name versus name+description labels for two GLiNER families across five languages, with per-language and per-model tuning, and (2) a multi-angle evaluation — exact-match and overlap F1, manual error inspection, and word-level SHAP with a paired ablation check — applied to a case where aggregate and example-level conclusions diverge.

2 Related Work

2.1 Zero-shot and few-shot NER

GLiNER reframes NER as matching text spans against natural-language label descriptions within a shared encoder, recognizing entity types never seen in training — unlike fixed-inventory pipelines such as spaCy or Stanza. Extensions include GLiNER2 [9], multi-task NER/classification/structured extraction, GLiNER multi-task [8], relation extraction, synthetic-data-augmented pre-training, and biomedical adaptation (GLiNER-BioMed). None of this prior work, to our knowledge, studies whether label phrasing itself — a bare name versus an expanded description — changes extraction quality, the question we isolate here.

2.2 Multilingual clinical NER and E3C corpus

Most clinical NER benchmarks are English-only, limiting cross-lingual generalization. The European Clinical Case Corpus [6] addresses this by annotating clinical case narratives in five European languages with clinical-entity, temporal, and factuality information. Closest to our use, [2] evaluate prompt-based and fine-tuned LLMs on MACCROBAT2020 and extend to further biomedical benchmarks including E3C, but under a single fixed labelling scheme rather than a name-vs-description comparison. Prior E3C work has largely focused on a single language or entity subtype; our use of all five languages under one generic "clinical entity" label treats language itself as an experimental variable alongside model family and label condition.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.114

Table 1: Document counts per split and language.

Language	Train	Validation	Test	Total
English	248	43	327	618
French	266	47	380	693
Italian	152	27	259	438
Spanish	238	43	422	703
Basque	475	84	719	1,278
Total	1,379	244	2,107	3,730

2.3 Explaining span-detection predictions with SHAP

SHAP (SHapley Additive exPlanations) attributes a model's output to its input features via a game-theoretic Shapley-value decomposition and is standard for explaining tabular and text classifiers [5]. Applying it to a span-detection model like GLiNER is less standard, since GLiNER has no fixed classification head — a prediction is the outcome of matching spans against a label description rather than a single output logit [4]. Our approach to adapting SHAP to this setting is described in Section 3.4.

3 Method

3.1 Dataset

We use the E3C clinical corpus in five languages (English, French, Italian, Spanish, Basque). Each languages origial train file is further split into a stratified train/validation set (85% / 15%, stratified by entity count per document); the original test file is left completely untouched until final scoring. Table 1 gives document counts. This three-way split was adopted after an earlier pipeline iteration reused one file for both tuning and reporting — a leakage risk discussed in Section 5. Here, the validation split is used exclusively for hyper-parameter tuning, and the test split is touched only once, at the end, to compute every number in Section 4.

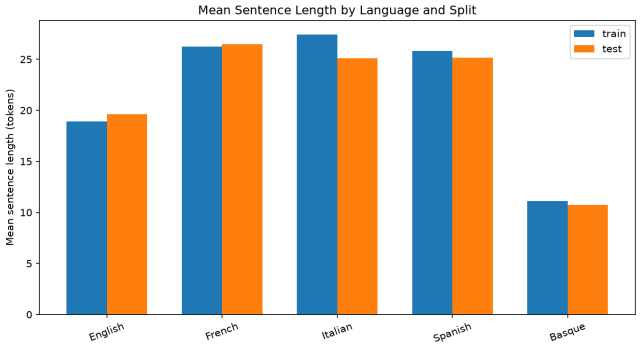


Figure 1: Mean sentence length in tokens for the train and test splits of each E3C language.

3.2 Model and label conditions

We fine-tune two model families: GLiNER1 (G1, small variant) and GLiNER2 (G2, base variant). For each model and language, we train under two label conditions: name (a short label, e.g., "clinical entity") and desc (the same name plus a longer natural language description of what qualifies as the entity — covering symptoms, diagnoses, and clinical findings and excluding, for instance, purely administrative or demographic mentions). Hyperparameters (learning rate, batch size, training steps/epochs, and, for G2, encoder/task learning-rate split and warm-up) are tuned per language and per model with Optuna [1] (20 trials, median pruning) against the validation split only; the test split is never used for tuning. We train final models with the tuned hyperparameters across three random seeds per language x condition to obtain a mean and standard deviation.

3.3 Evaluation

We report exact-match F1 (spans must coincide exactly, IoU = 1.0) and overlap F1 (IoU $\geq$ 0.5,), the latter giving partial credit for spans in roughly the right place but slightly mis-bounded. Both metrics use the same greedy one-to-one matching in descending order of IoU, so a single predicted span cannot be credited against more than one gold span. All numbers are computed on the held-out test split only, after confirming that reloaded checkpoints reproduce their training-time scores.

3.4 Confidence-threshold sweep

GLiNER assigns each predicted span a confidence score; the F1 numbers in Section 4 use a single fixed threshold. To see how each model behaves away from that point, we reuse each language×condition's first-seed checkpoint (no retraining) to sweep the threshold from 0.3 to 0.8 and recompute precision, recall, and F1. This is diagnostic rather than a headline result: it shows where the precision/recall trade-off sits for a given language and model — information a single-threshold F1 cannot convey — useful for choosing between a high-recall screening use and a high-precision autofill use. Full per-language curves are kept in supplementary logs.

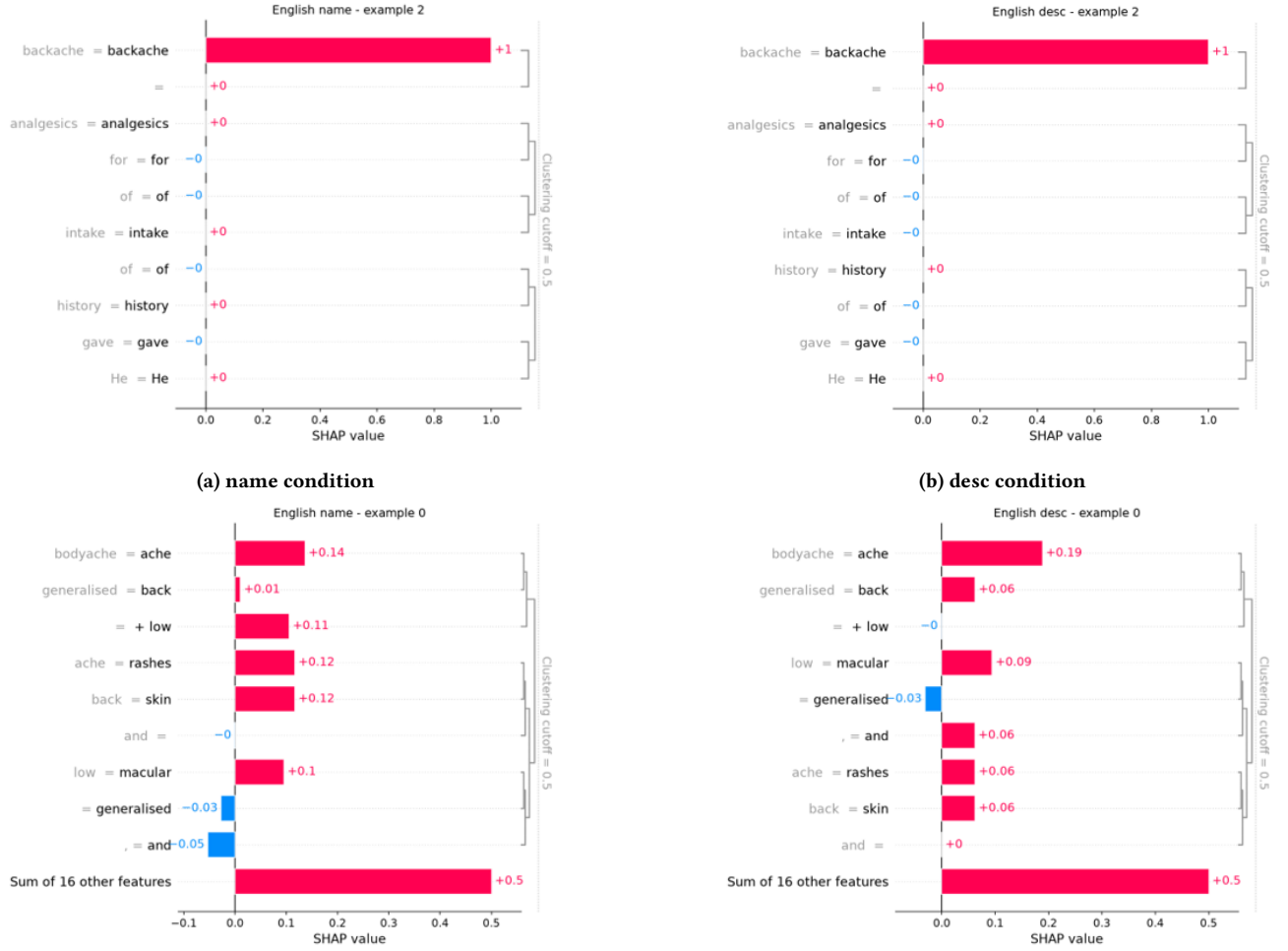
3.5 Qualitative error inspection and interpretability (SHAP)

Alongside aggregate metrics, we manually inspect false positives and false negatives on real test documents for both label conditions and model families; this surfaced the concrete divergence analysed in Section 4.3, where an aggregate score moves one way while a clinically salient entity moves the other. To probe individual predictions further, we define a scoring function equal to the model's single most-confident predicted span for the target label and apply SHAP to it, yielding a signed per-word attribution (positive = pushes toward detection, negative = pushes away). As a sanity check beyond the additive Shapley approximation, we run a paired before/after ablation: removing the single highest-attribution word and re-running the model, to confirm the attribution reflects a genuinely load-bearing word rather than a masking artifact.

4 Results

4.1 Exact-match and overlap F1 (test set)

Table 2 reports exact-match F1 and Table 3 reports overlap F1 across all language x model x condition combinations. Overlap F1 exceeds exact F1 for every language, model, and condition (e.g. English G1-name: 0.861 overlap vs. 0.765 exact), confirming that most misses are near-misses on span boundaries rather than missed entities entirely; G2's overlap-exact gap (7–12 points) is consistently larger than G1's (8–9 points), meaning G2 finds roughly the right region slightly more often than it gets the exact boundary. Across all ten language x condition pairs, G1 wins



(a) name condition **(b) desc condition**

(c) name condition **(d) desc condition**

Figure 2: Word-level SHAP attributions for two further correctly-predicted English test sentences under both label conditions. Leftmost pair: nearly all context words carry near-zero attribution under both conditions, and the target entity word alone (*backache*) drives the prediction (+1), indicating a case where the model's decision is robust to the choice of label description. Rightmost pair: attributions are decomposed into individual clustered word-pair contributions plus the aggregated contribution of the remaining 16 lower-ranked features; the dominant contributor shifts from *low* → *ache* (+0.25 under name) to *bodyache* → *ache* (+0.19 under desc), and the sign of *generalised* → *skin/back* flips between conditions, illustrating that the two label conditions can rely on different words even when both correctly detect the entity.

Table 2: Exact-match F1, mean ± std. across 3 seeds, held-out test set. Higher of G1/G2 per row shown for reference in text.

Language	Cond.	G1 F1 (±sd)	G2 F1 (±sd)
English	name	0.7654 ± .0025	0.7283 ± .0116
English	desc	0.7616 ± .0021	0.7322 ± .0069
French	name	0.6337 ± .0075	0.6426 ± .0106
French	desc	0.6297 ± .0049	0.6386 ± .0077
Italian	name	0.6479 ± .0052	0.6778 ± .0026
Italian	desc	0.6658 ± .0078	0.6776 ± .0053
Spanish	name	0.6616 ± .0082	0.6410 ± .0034
Spanish	desc	0.6499 ± .0031	0.6462 ± .0023
Basque	name	0.7127 ± .0066	0.7043 ± .0063
Basque	desc	0.7265 ± .0021	0.7121 ± .0037

exact-match F1 in 6/10 (English x2, Spanish x2, Basque x2) and G2 wins in 4/10 (French x2, Italian x2); on overlap F1, G1 wins 9/10, losing only a near-tie on Italian-name. G1 is therefore the stronger and more consistent model in this setting, with G2's advantage concentrated in French and Italian — a difference that may partly reflect the small/base size mismatch between the two model families rather than the fine-tuning process alone.

4.2 Effect of the label description

The description's effect is small and inconsistent. For G1, name beats desc in all five languages by a small, consistent margin. For G2, it flips by language: desc wins for English and Spanish, name wins for French, Italian, and Basque — so the answer depends on model family, not just language.

Table 3: Overlap F1 (IoU ≥ 0.5), held-out test set.

Language	Cond.	G1 overlap F1	G2 overlap F1
English	name	0.8610	0.8152
English	desc	0.8510	0.8173
French	name	0.7550	0.7323
French	desc	0.7614	0.7308
Italian	name	0.7451	0.7415
Italian	desc	0.7587	0.7415
Spanish	name	0.8050	0.7282
Spanish	desc	0.7940	0.7292
Basque	name	0.8131	0.7747
Basque	desc	0.8316	0.7800

Table 4: Same sentence, four model×condition combinations.

Config.	Predicted entities	Result
G1 —- name	fever, backache, splenomegaly, Brucellosis, skin rash	5/6, misses gi bleed
G1 —- desc	fever, backache, splenomegaly, Brucellosis, skin rash	5/6, misses gi bleed
G2 —- name	fever, skin rash, backache, splenomegaly, Brucellosis, gi bleed	6/6, perfect
G2 —- desc	fever, backache, skin rash, splenomegaly, Brucellosis, “infection”	misses gi bleed + 1 false positive

4.3 SHAP case study

For the English test sentence "Residing in Middle East with background history of fever, backache, skin rash, splenomegaly and a low platelet count possibility of atypical infection like Brucellosis with NSAID induced gi bleed was considered," gold contains six entities including "gi bleed" (Table 4). Only G2 with the short name recovers all six; G1 (both conditions) and G2-desc all miss "gi bleed," and G2-desc additionally hallucinates "infection." Notably, G2-desc wins on aggregate English F1 (0.7322 vs. 0.7283) — a small aggregate gain can still lose on individual, clinically salient examples.

At the word level, SHAP attributions are intuitive: for a separate sentence, "rashes" carries the largest positive attribution (+0.33) while the negator "non" (as in "non pruritic") pushes negatively (−0.11). Sentences carried by several moderate positive bars appear more robust to small wording changes than single-dominant-bar cases. A paired ablation removing "rashes" confirms the direction: detection confidence drops, matching the SHAP attribution.

5 Discussion and Limitations

Three takeaways follow from Section 4. Model-family choice matters more than label engineering: the G1/G2 gap (up to 5–6 F1 points on English and Spanish) exceeds the name/desc gap within either model. A longer description is not a safe default — it helped G2 on two of five languages but never G1. And aggregate F1 can mask example-level failures that matter clinically, as the "gi bleed" case shows; word-level SHAP is a practical way to audit for this. Limitations: G1 and G2 are not size-matched, so part of the gap may reflect scale rather than architecture; Italian's small validation set (27 documents) meant validation-time ranking did not match final test ranking, so its conclusions should be treated cautiously; SHAP proxies detection confidence, not boundary correctness; and the threshold sweep uses a single seed's checkpoint, so its operating points are illustrative. Finally, this study uses a single generic "clinical entity" label rather than fine-grained subtypes — testing whether the label-length trade-off changes for narrower entity types is a natural next step.

6 Conclusion

Across five languages, GLiNER1 is the more consistent clinical NER model, with GLiNER2's advantage concentrated in French and Italian; a descriptive label helps GLiNER2 on some languages but never GLiNER1. Overlap-based evaluation and word-level SHAP add information exact-match F1 alone does not — most errors are near-miss boundaries, and SHAP shows concretely which words drive or undermine individual predictions, as the "gi bleed" case illustrates. These results argue for evaluating zero-shot NER label choices with more than a single aggregate F1 number. Future work includes fine-grained entity subtypes, additional model families under the same leakage-free protocol, and a human evaluation of whether SHAP explanations help clinical annotators audit predictions faster than raw output alone.

Acknowledgements

This work was supported by the Slovenian Research Agency. Funded by the European Union. UK participants in Horizon Europe Project PREPARE are supported by UKRI grant number 10086219 (Trilateral Research). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA) or UKRI. Neither the European Union nor the granting authority nor UKRI can be held responsible for them. Grant Agreement 101080288 PREPARE HORIZON-HLTH-2022-TOOL-12-01.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

[2] Erik Calcina, Erik Novak, Dunja Mladenić, and PREPARE Project Group. 2026. Extracting biomedical entities from clinical records: a comparative study of large language model approaches. *Applied Artificial Intelligence*, 40, 1, 2700905.

[3] Lenart Dolinar, Erik Calcina, and Erik Novak. 2025. Evaluating open-source large language models for synthetic non-english medical data generation using prompt-based techniques. *Informatica*, 49, 27, 125–134.

[4] Xueguang Li, Di Lin, Xue Jiang, Yanxi Li, and Yugang Chi. 2026. Explanation-guided medical named entity recognition with stability and boundary awareness for atopic dermatitis. *arXiv preprint arXiv:2606.22886*.

[5] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[6] Bernardo Magnini, Begoña Altuna, Alberto Lavelli, Manuela Speranza, and Roberto Zanoli. 2020. The e3c project: collection and annotation of a multilingual corpus of clinical cases. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*, 190–196.

[7] Abdul Sittar, Mateja Smiljanic, Alenka Guček, and Marko Grobelnik. 2026. Fake news detection through llm-driven text augmentation across media and languages. *Machine Learning and Knowledge Extraction*, 8, 4, 103.

[8] Ihor Stepanov and Mykhailo Shtopko. 2024. Gliner multi-task: generalist lightweight model for various information extraction tasks. *arXiv preprint arXiv:2406.12925*.

[9] Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. 2025. Gliner2: schema-driven multi-task learning for structured information extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 130–140.

[10] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. Gliner: generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, 5364–5376.

Data Efficiency of GLiNER: A Multilingual and Multi-Scale Learning Curve Study

Nina Kokalj
Jožef Stefan Institute
Ljubljana, Slovenia
nina.kokalj123@gmail.com

Erik Calcina
Jožef Stefan Institute
Ljubljana, Slovenia
erik.calcina@ijs.si

Abdul Sittar
Jožef Stefan Institute
Ljubljana, Slovenia
abdul.sittar@ijs.si

Erik Novak
Jožef Stefan Institute
Ljubljana, Slovenia
erik.novak@ijs.si

Abstract

Manually annotated clinical data are expensive to obtain, making data-efficient adaptation of named entity recognition models particularly important. We study how GLiNER performance changes with increasing amounts of task-specific training data across three model sizes, four clinical datasets, and multiple languages. Models were fine-tuned on nested subsets containing 10%, 30%, 60%, 80%, and 100% of the available training data and evaluated across three training seeds. The largest gains occurred early: with only 10% of the training data, models already captured approximately 52–76% of the total improvement from zero-shot to full-data performance, increasing to 88–97% at 60%. Larger models often provided the greatest advantage in low-data settings, although this was not consistent across datasets. Learning dynamics also varied across languages, while persistent span-boundary errors remained even after overall performance began to plateau. These results highlight the potential of learning curves to guide annotation effort in iterative clinical NER workflows.

Keywords

named entity recognition, gliner, learning curves, data efficiency

1 Introduction

Named Entity Recognition (NER) is a key component of many clinical text processing pipelines, but adapting NER systems to a specific domain, language, or annotation scheme typically requires manually annotated data. In practice, such annotations are costly to obtain, making it important to understand how much training data is actually needed to achieve useful performance. This question is particularly relevant for models such as GLiNER, which support flexible entity schemas and can be fine-tuned for task-specific extraction [7].

This work was motivated by a human-in-the-loop workflow developed within the European PREPARE project. An initial model identifies entities in clinical documents, which users review and correct. The corrected examples are added to the training data, and the model is retrained as more annotated documents become available. Ideally, this progressively reduces the amount of manual correction required. However, it remains unclear how quickly these gains occur and whether they differ across model sizes and experimental settings.

We evaluate three GLiNER model sizes across five nested training-data fractions, multiple seeds, datasets, and languages to compare how performance evolves with increasing annotation.

2 Related Work

Recent approaches to NER aim to combine the flexibility of large language models with the efficiency of encoder-based architectures. Zaratiana et al. introduced **GLiNER**, a compact bidirectional transformer for flexible NER using natural-language entity labels [7]. While GLiNER showed strong zero-shot performance across multiple benchmarks, the original work did not systematically examine how performance changes with the amount of task-specific fine-tuning data.

In clinical NER, Kormilitzin et al. introduced Med7 for extracting medication-related concepts from electronic health records and showed that performance dropped when the model was transferred to a different clinical setting, but improved substantially after fine-tuning on a small amount of target-domain data [3].

The amount of annotated data available for fine-tuning is particularly important in low-resource and domain-specific NER. Seeha et al. investigated this effect for German medical NER by fine-tuning BERT- and LLaMA-based models on increasing fractions of two annotated datasets, observing large initial gains followed by more gradual improvements as additional data were introduced [5]. More broadly, Urbizu et al. studied how model size and training data interact in low-resource settings using three BERT model sizes across four languages. They found that scaling behavior in such settings differs from that observed for large language models, with smaller-scale models generally requiring relatively more training data to achieve efficient performance [6]. While previous work has examined training-data requirements in NER and scaling in BERT-based models, GLiNER learning curves across model sizes remain underexplored.

3 Methodology

3.1 Dataset characteristics

The experiments were conducted on four clinical NER datasets with different characteristics. **SMMN (Synthetic Multi-Med Notes)** is a multilingual synthetic clinical NER dataset released by E3-JSI [2]. **E3C (European Clinical Case Corpus)** is a multilingual corpus of clinical case narratives collected from heterogeneous, freely licensed sources [4]. **AHR-NL and AHR-IT (Annotated Health Records in Dutch and Italian; study-specific names)** are non-public de-identified clinical datasets

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.118

provided by project partners, originating from hospital settings. In the present experiments, E3C uses a single broad clinical entity label, whereas SMMN, AHR-NL, and AHR-IT contain multiple entity types. Table 1 summarizes the main dataset characteristics.

Table 1: Dataset characteristics and size of the full training partition. For multilingual datasets, sizes are reported as the minimum–maximum range across languages.

Dataset	Lang.	Source	Entities	Docs
SMMN	9	Synthetic	Multiple	447-582
E3C	5	Clinical cases	Single broad	798-816
AHR-NL	NL	Hospital	Multiple	488
AHR-IT	IT	Hospital	Multiple	1623

3.2 Data preparation

For each dataset-language combination, the original documents were divided into **training (72%), validation (8%), and test (20%) sets** using stratified splitting to preserve the entity-label distribution across subsets. Long documents were chunked after splitting at approximately 384 tokens while preserving sentence and entity boundaries whenever possible. The validation and test sets were fixed throughout all subsequent experiments.

3.3 Hyperparameter optimization

Before the learning-curve experiment, hyperparameters were optimized separately for each dataset–language–model combination using **60 Optuna trials** [1] on the full training partition. Models were trained on the training set and evaluated exclusively on the fixed validation set. The best-performing hyperparameter configuration was then used for the subsequent learning-curve experiments.

3.4 Learning-curve experiment

To evaluate the effect of training data size, we created five **nested, stratified subsets** containing **10%, 30%, 60%, 80%, and 100%** of the training documents. We denote the subset containing x% of the training data as D_x, such that $D_{10} \subset D_{30} \subset D_{60} \subset D_{80} \subset D_{100}$. Each larger subset retained all documents from the preceding fraction.

For each fraction, models were trained using the selected hyperparameters. Prediction thresholds were optimized on the fixed validation set by maximizing exact micro-F1 and then applied once to the fixed test set. Experiments were repeated with **three seeds (42, 43, and 44)** using identical training subsets. The zero-shot baseline used no task-specific fine-tuning, with its threshold calibrated on the same validation set.

We evaluated three GLiNER model sizes: `gliner_small-v2`[1], `gliner_medium-v2`[2], and `gliner_large-v2`[3] [7]. Performance was measured using exact entity-level micro-averaged precision, recall, and F1. We also calculated relaxed F1, counting predictions as correct when they overlapped the gold span by at least one token; the relaxed–exact F1 difference is referred to as the *boundary gap*.

For cross-dataset summaries, seeds were averaged within each language–model configuration and configurations were weighted equally. Language-specific curves were averaged equally across model sizes after seed averaging. Seed variability was summarized by the sample SD of the three seed-specific dataset-level mean exact micro-F1 scores. Cumulative gain was computed as $(F1_f - F1_0)/(F1_{100} - F1_0)$, where $F1_f$ is the exact micro-F1 at training fraction f, $F1_0$ the zero-shot score, and $F1_{100}$ the full-data score. Marginal gain was the F1 increase between consecutive fractions, normalized per additional 10% of training data. For AHR-IT, the large model was excluded from aggregate learning-curve analyses due to high seed instability.

[1] https://huggingface.co/urchade/gliner_small-v2
[2] https://huggingface.co/urchade/gliner_medium-v2
[3] https://huggingface.co/urchade/gliner_large-v2

4 Results

4.1 Early gains and diminishing returns

Across all four datasets, the largest performance improvement occurred after introducing the first supervised training fraction, followed by progressively smaller gains as more training data were added. Figure 1a shows the cumulative share of the total exact micro-F1 improvement between the zero-shot baseline and training on 100% of the available data. The corresponding mean exact micro-F1 (zero-shot→full-data) was 0.144→0.624 for SMMN, 0.339→0.713 for E3C, 0.032→0.457 for AHR-NL, and 0.190→0.662 for AHR-IT. With only 10% of the training data, the models already captured approximately **52–76%** of this total improvement. At 30%, this increased to approximately **78–91%**, and by 60% all datasets had reached roughly **88–97%** of the improvement observed at full data.

The same pattern is visible in the marginal gains in Figure 1b. The transition from zero-shot inference to training on the first 10% yielded an improvement of approximately 22–37 exact-F1 points across datasets. Increasing the training fraction from 10% to 30% provided only about 3.5–5.6 F1 points per additional 10% of training data, while later increments generally contributed less than 2 points per additional 10%. SMMN approached its full-data performance particularly early, whereas AHR-NL retained comparatively larger gains at later fractions. AHR-IT showed minor non-monotonic variation between 60% and 80%. Across the aggregate learning curves, seed variability was generally low (median SD = 0.004; 90th percentile = 0.015).

4.2 Model-size effects across training fractions

Figure 2 compares the exact micro-F1 of the large and small GLiNER models across training fractions. For SMMN, E3C, and AHR-IT, the largest advantage of the large model occurred at 10% of the training data, reaching +2.4, +8.8, and +16.9 F1 points, respectively. At 30%, these differences decreased to +1.5, +4.6, and +7.9 points. At higher fractions, the gap remained small for SMMN and considerably below its 10% value for E3C.

The pattern was different for AHR-NL, where the small model outperformed the large model at four of the five fractions, including by 1.7 F1 points at full data. AHR-IT also showed substantial variation at larger fractions. At 100%, the large AHR-IT model ranged from 0.274 to 0.652 exact micro-F1 across seeds; its apparent 20.9-point disadvantage should therefore be interpreted cautiously. Overall, larger models tended to benefit most when supervision was limited, while increasing training data reduced the performance gap between model sizes. At full data, large-model training was **1.5–4.4**× slower than small-model training in our setup.

4.3 Cross-language learning dynamics

The two multilingual datasets showed different language-specific learning trajectories (Figure 3). In SMMN, the nine languages

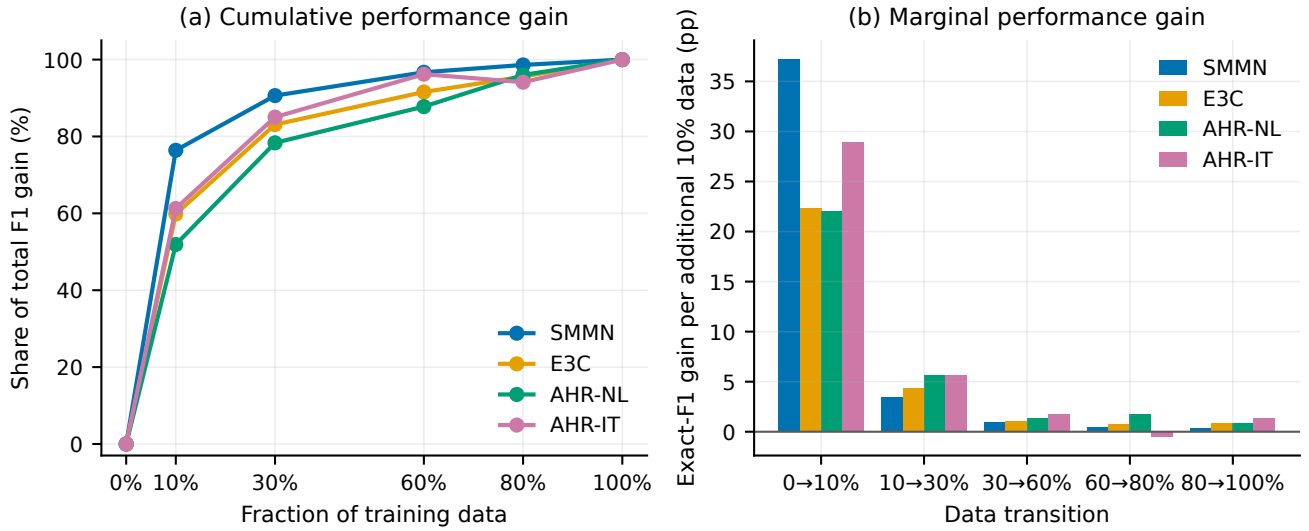


Figure 1: Cumulative share of the total zero-shot-to-100% exact micro-F1 gain (a) and marginal gain per additional 10% of training data (b), measured in percentage points (pp). AHR-IT aggregates include only the stable small and medium models.

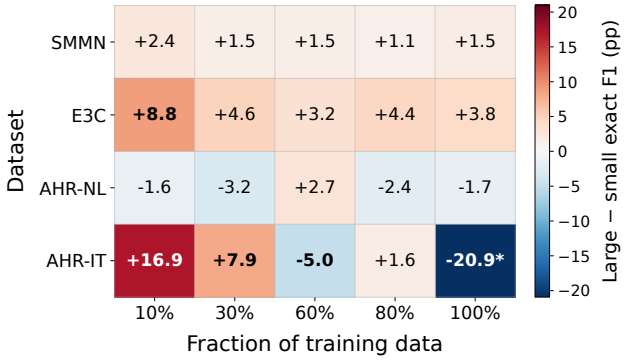


Figure 2: Exact micro-F1 difference between the large and small GLiNER models across training fractions. Positive values favour the large model. *AHR-IT large at 100% showed high seed variability (SD = 0.191).

followed similar curves and were already closely clustered after training on 10% of the data. The remaining differences decreased further as additional supervision was introduced.

E3C showed greater cross-language variation at low training fractions, followed by stronger convergence as more data became available. Basque provides the clearest example: it started well below the other languages in the zero-shot and 10% settings but improved substantially with additional supervision, eventually reaching a similar performance range. English followed a different trajectory, starting from a much stronger baseline and showing smaller subsequent gains.

4.4 Global plateaus mask persistent boundary errors

The gap between relaxed and exact-F1 remained substantial even at high training fractions for some datasets. The mean boundary gap decreased only from approximately 11.9 F1 points at 10% to 10.7 at 100% in SMMN and from 16.3 to 14.3 points in E3C. At full data, the corresponding gaps were smaller for the stable AHR-IT models and AHR-NL, at approximately 4.9 and 3.1 points, respectively.

Label-level results from SMMN further showed that similar exact-F1 performance can reflect different error profiles. *Observation*, the consistently most difficult entity type, reached only about 0.293 exact-F1 at full data but had a boundary gap of approximately 2.1 points. In contrast, *Drug dose*, *Event*, *Rehabilitation*, and *Operation* showed much larger gaps, ranging from approximately 22.7 to 29.9 F1 points. For these entity types, a considerably larger share of the exact-match error was associated with span-boundary mismatches.

5 Discussion

5.1 Annotation and model trade-offs

For iterative annotation workflows, our results suggest that **annotation provides its highest return early in the adaptation process**. A model trained after an initial annotation round can already capture a large share of the improvement achievable with the full training set. However, the point at which further annotation becomes less beneficial is dataset-dependent. The results therefore do not support a single stopping fraction across datasets.

Model selection should similarly depend on the available annotation budget. Where a large-model advantage was present, it was concentrated in the low-data regime, while smaller models generally became more competitive as supervision increased. This pattern was not universal, however. AHR-NL showed no consistent benefit from increased model size, and the large AHR-IT model also exhibited substantial instability at high training fractions. Smaller models may therefore offer a better performance–training-time trade-off with more training data.

5.2 Dataset and language effects

The multilingual results show that the same annotation fraction can yield different gains across languages. This contrast may partly reflect the more homogeneous synthetic construction of SMMN versus the heterogeneous clinical content of E3C, although causality is not established.

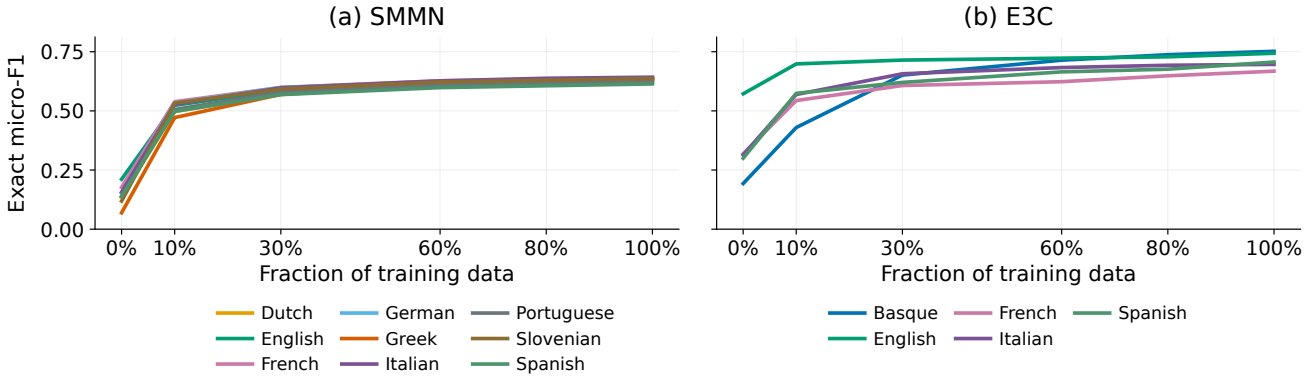


Figure 3: Cross-language exact micro-F1 learning curves for the multilingual SMMN and E3C datasets.

The E3C results suggest that a uniform multilingual annotation budget may be inefficient. Stronger languages may reach diminishing returns earlier, while weaker ones can continue to benefit from additional supervision. Annotation effort could therefore be guided by language-specific learning progress rather than fixed fractions.

5.3 Implications for targeted annotation

Once overall performance begins to plateau, the remaining errors may provide a more useful guide for annotation than the global F1 score alone. Later annotation rounds could therefore shift from broad data expansion toward more targeted examples, prioritizing weaker languages, difficult entity types, or recurring span-boundary errors. Entity types with large boundary gaps may benefit from examples that clarify span boundaries, whereas labels with low exact-F1 but small boundary gaps may require greater focus on recognition or classification errors. Although our experiments do not directly evaluate such a targeted strategy, the results suggest that uniformly adding more data may become increasingly inefficient as the learning curve flattens.

6 Limitations

Several limitations should be considered when interpreting the results. First, the training fractions represent **dataset-specific proportions rather than equal absolute annotation budgets**. A 10% subset may therefore contain different numbers of documents and annotated entities across datasets and languages. The observed saturation points should consequently not be interpreted as a universal amount of training data required by GLiNER.

Second, each configuration was repeated with three training seeds, which limits the precision with which training variability can be estimated. This is particularly relevant for unstable configurations such as the large AHR-IT model.

Finally, hyperparameters were optimized before the learning-curve experiment and then kept fixed across training fractions. This enables a consistent comparison of training-set size, but the optimal configuration may itself depend on the amount of available training data. Future work could examine fraction-specific optimization and evaluate whether the observed learning patterns generalize to additional datasets and annotation settings.

7 Conclusion

The largest gains occurred at low training fractions, followed by diminishing returns. Larger models tended to perform better in the low-data regime, while their advantage generally narrowed as more training data became available. Learning dynamics also varied across datasets and languages, while span-boundary errors persisted after overall performance began to plateau. These findings suggest that annotation effort should follow observed learning progress, with later rounds targeting weaker languages, difficult entity types, and persistent error patterns.

Acknowledgements

This work was supported by the Slovenian Research Agency. Funded by the European Union. UK participants in Horizon Europe Project PREPARE are supported by UKRI grant number 10086219 (Trilateral Research). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA) or UKRI. Neither the European Union nor the granting authority nor UKRI can be held responsible for them. Grant Agreement 101080288 PREPARE HORIZON-HLTH-2022-TOOL-12-01.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2623–2631. doi:10.1145/3292500.3330701.

[2] E3-JSI. 2026. Synthetic multi-med notes for ner. Hugging Face. (2026). https://huggingface.co/datasets/E3-JSI/synthetic-multi-med-notes-ner-v1.

[3] Andrey Kormilitzin, Nemanja Vaci, Qiang Liu, and Alejo Nevado-Holgado. 2021. Med7: a transferable clinical natural language processing model for electronic health records. *Artificial Intelligence in Medicine*, 118, 102086. doi:10.1016/j.artmed.2021.102086.

[4] Bernardo Magnini, Begoña Altuna, Alberto Lavelli, Anne-Lyse Minard, Manuela Speranza, and Roberto Zanoli. 2023. European clinical case corpus. In *European Language Grid*. Springer, Cham, 283–288. doi:10.1007/978-3-031-17258-8_17.

[5] Suteera Seeha, Sihan Wu, Justin Hofenbitzer, Claudio Benzoni, Peter Pallaoro, Raphael Scheible, Martin Boeker, and Luise Modersohn. 2025. German medical NER with BERT and LLMs: the impact of training data size. In *Intelligent Health Systems – From Technology to Data and Knowledge, Proceedings of MIE 2025* (Studies in Health Technology and Informatics). Vol. 327. IOS Press, (May 2025), 798–802. ISBN: 9781643685960. doi:10.3233/SHTI250469.

[6] Gorka Urbizu, Iñaki San Vicente, Xabier Saralegi, Rodrigo Agerri, and Aitor Soroa. 2023. Scaling laws for BERT in low-resource settings. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, (July 2023), 7771–7789. doi:10.18653/v1/2023.findings-acl.492.

[7] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Association for Computational Linguistics, Mexico City, Mexico, (June 2024), 5364–5376. doi:10.18653/v1/2024.naacl-long.300.

Comparing Pseudo-Labeling Criteria for Active Learning on a Service-Desk Dataset

Rok Klančič
Jožef Stefan Institute
Ljubljana, Slovenia
rok.klancic@gmail.com

Klemen Kenda
Jožef Stefan Institute
Ljubljana, Slovenia
klemen.kenda@ijs.si

Abstract

Recent advances in large language models and human-machine interfaces have made it increasingly feasible to automate aspects of IT support ticket management. Since manual ticket routing can be slow and error-prone, ticket-routing models can help mitigate these issues. Training such models, however, requires organisations to label IT support tickets with the team responsible for handling them. This labelling is expensive because it requires expert knowledge, the tickets are unevenly distributed among the specialised teams and ticket content is sensitive. We study pseudo-labeling criteria that, during a simulated active-learning process, decide which unlabelled tickets the model can label itself. We compare nine criteria, including raw and calibrated confidence, k-nearest-neighbour agreement with and without calibration, SVM decision margins, outlier rejection, class-conditional thresholds and ensemble agreement, at checkpoints of 100, 250 and 500 labelled tickets. The experiments use a real IT support ticket dataset. With about half of the tickets labelled, the best-performing criteria label the entire remaining pool with around 97% accuracy. Agreement-based criteria are the most dependable throughout, whereas post hoc calibration collapses below roughly a quarter of the labels and becomes competitive with the best agreement- and margin-based criteria once the underlying model is sufficiently accurate.

Keywords

active learning, pseudo-labeling, IT support ticket routing, confidence calibration

1 Introduction

Many organisations operate internal service desks where incoming tickets must be routed to the support team best able to resolve them. When a ticket is sent to the wrong queue, it must be reassigned manually and often takes longer to resolve. An accurate automatic router would therefore help mitigate this issue [1]. Building such a router with supervised methods, however, requires a collection of tickets labelled with their handling team. This labelling is expensive for several reasons: experts must understand the responsibilities of many teams, the distribution of team labels is long-tailed with many rare classes, and the ticket content is sensitive. Labelling therefore cannot be outsourced and must be performed by the organisation's own staff.

Active learning (AL) can reduce the number of labels required by repeatedly selecting the tickets that are most informative to the current model [6, 9]. It does not, however, remove the human from the loop: every selected ticket still has to be labelled by an expert. Our initial experiments on service-desk dataset indicate that this effort is concentrated on the most difficult tickets. During an AL run using uncertainty sampling [5], the model predicted the correct team for only about 30% of the queried tickets before their labels were revealed, compared with about 80% using random sampling. Consequently, a labelling process leaves behind a pool of unlabelled tickets that, by construction, the model already finds easier.

This paper asks how much of the remaining labelling work the model can perform itself. We study pseudo-labeling [4] criteria that decide, for each unlabelled ticket, whether the model's prediction is sufficiently trustworthy to be accepted as a label, so that the tickets that pass can be labelled automatically while the remainder continues in the AL process. We compare nine such criteria in an active-learning simulation on an IT support ticket dataset (Section 4) and ask which criterion labels tickets most accurately and how much labelling effort it saves at that accuracy.

Our contribution is a systematic evaluation of these criteria on a real service-desk dataset in the low-label regime that a labelling process actually faces.

2 Related Work

2.1 Active Learning

AL reduces annotation costs by allowing a learning algorithm to select the most informative instances. In the common pool-based setting, an unlabelled pool is available and a query strategy iteratively selects instances for labelling by an oracle, typically a human annotator [6]. Strategies include uncertainty sampling [5], query-by-committee, expected model change, variance reduction, estimated error reduction and density-weighted methods [6]. Their common goal is to achieve competitive predictive performance with substantially fewer labelled examples. Surveys do not identify a single best strategy, however the best practical recommendation is to combine informativeness (e.g. uncertainty) with representativeness (e.g. density) strategies. They also note that AL is the most effective when labels are expensive, initial labeled set is sufficient and the model's uncertainty estimates are reliable [6, 9].

2.2 Pseudo-Labeling and Self-Training

Pseudo-labeling [4] and self-training instead exploit unlabelled data by assigning labels to examples for which the model's prediction is considered sufficiently reliable. FixMatch, for example, retains only high-confidence pseudo-labels and uses them as training targets [8]. FlexMatch extends this approach with class-dependent, dynamically adjusted confidence thresholds to account for differences in class difficulty [11]. More directly related to AL, in [3] confidence-based AL is combined with self-training to reduce the amount of data requiring human annotation. AdaPL integrates adaptive pseudo-label selection into deep AL, setting

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.117

thresholds according to the uncertainty of labelled and unlabelled data [2]. Together, these approaches show the potential of combining human and automatic labelling, while highlighting the need to identify trustworthy predictions reliably.

2.3 Confidence Calibration and Agreement

The reliability of model confidence is especially important when confidence scores determine which examples are labelled automatically. Raw classifier confidence do not necessarily correspond to the empirical probability of correctness. Post hoc methods such as Platt scaling and isotonic regression can produce more reliable probability estimates [7]. Research on pseudo-labeling likewise shows that better-calibrated confidence can produce more reliable pseudo-labels [10]. Agreement between multiple predictors offers a complementary measure of label reliability: Tri-training, for example, uses agreement between independently trained classifiers to label previously unlabelled instances [12].

2.4 Positioning of This Work

Our work brings these lines of research together by studying how the remaining unlabelled pool can be labelled automatically during active learning. We evaluate raw and calibrated confidence, classifier margins, neighbourhood agreement, combinations of confidence and neighbourhood information, ensemble agreement, outlier rejection, and class-conditional criteria. Unlike approaches that use pseudo-labels primarily as an additional training signal, we focus on the reliability of automatically assigned labels and on how it changes as AL acquires additional human-labelled instances. What remains insufficiently addressed by prior work is precisely this comparison: no systematic study evaluates the reliability of such labelling criteria across the stages of an AL process, which is the gap this paper fills.

3 Methodology

The central object of our study is a pseudo-labeling criterion. Given a classifier fitted on the labelled tickets and the unlabelled pool, the criterion decides which tickets can be labelled automatically and which must be left to human annotators. We compare nine such criteria within the process described in Section 4.

3.1 Pseudo-Labeling Criteria

3.1.1 Confidence and Neighbourhood Criteria. The first three criteria threshold the model's confidence: raw maximum posterior probability, maximum posterior after Platt scaling, and maximum posterior after isotonic recalibration. Both calibrators are fitted on the held-out split. A fourth criterion uses k-nearest-neighbour agreement instead of probabilities: a ticket is accepted when at least $k-1$ of its k nearest labelled neighbours, measured by cosine distance in the full feature space of the MiniLM text embedding concatenated with the service and subcategory indicators, share the model's predicted label. We sweep this criterion over $k \in \{3, 5, 7\}$ as simple defaults, with the accept rule set to $k-1$ or k, and report the best operating point of the sweep. The fifth criterion requires both Platt confidence above the threshold and neighbour agreement with the central setting $k = 5$ (four of five agreeing), so that only its confidence threshold is swept.

3.1.2 Margins and Outlier Rejection. The SVM margin criterion uses the distance to the one-vs-rest decision boundary of the predicted class rather than a probability. Its thresholds are quantiles of the margins on the held-out split. The outlier-rejection criterion combines a confidence threshold with an Isolation Forest (contamination 0.05) fitted on the training features, so that only in-distribution tickets are labelled.

3.1.3 Class and Ensemble Criteria. The class-conditional criterion selects a separate confidence threshold for each class on the held-out split, targeting a class precision of 90%, 95% or 99%. Classes with too little support are never automatically labelled. The ensemble-agreement criterion labels a ticket only when all three models (Section 3.2) predict the same class and their average confidence exceeds the threshold.

3.2 Models

All criteria use the same three classifiers: logistic regression ($C = 10$), a support vector machine with an RBF kernel ($C = 10$, with probability estimates enabled), and a random forest with 200 trees. These values are simple defaults chosen a priori rather than the result of a systematic tuning study; because every criterion shares the same frozen models at each checkpoint, however, their exact values shift all criteria jointly and cannot affect the comparison between criteria, and absolute operating points under other settings may differ. Each model is refit on all labelled tickets at every labelling step. At a given checkpoint, all criteria use the same frozen models, so differences between criteria cannot be attributed to different underlying models. The ensemble-agreement criterion combines all three.

4 Experiment Setting

4.1 Dataset

We use anonymised IT support tickets collected and provided by GFT. Each ticket contains a title and description, the relevant service and subcategory, the conversation log, and the team that resolved it. The task is to predict the handling team from the text and metadata of an incoming ticket. After preprocessing, the dataset contains 1,172 tickets labelled with 21 teams.

4.2 Data Preprocessing

We apply three filters. First, we remove tickets reassigned to another team before their final team, leaving each ticket with one unambiguous label. Second, we discard teams with fewer than ten tickets because they provide too few examples for efficient learning. Third, we drop tickets missing a title and description or a team label. Each feature vector concatenates a 384-dimensional all-MiniLM-L6-v2 embedding of the title and description with one-hot indicators for the service and subcategory. A stratified 80/20 split produces 937 training tickets and 235 held-out tickets. The training split is the AL pool, while the held-out split is used for calibration when required.

4.3 Active Learning Process

The process starts with no labelled tickets, so its first tickets are selected by the query strategy rather than a manually constructed seed set. At this point the classifier cannot be fitted and produces uniform probabilities, so the first least-confidence queries are tied and resolved by random tie-breaking at a fixed seed; i.e. the process starts from a small effectively random seed. Unlabelled tickets from the training split are queried in batches of two, labelled with their true team, and the models are refit on all labelled data after each batch. We compare random sampling, which selects tickets uniformly from the unlabelled pool,

Table 1: Best-achievable (retrospective) operating points: Coverage (Accuracy) at the configuration with the highest coverage subject to accuracy ≥ 0.90, per checkpoint and query strategy, averaged over the three base classifiers. † indicates that the accuracy target is unattainable at that checkpoint and that the best-accuracy result is reported instead. A dash indicates that the criterion produced no labels. SVM margin* operates on the SVM only.

	Random sampling			Uncertainty sampling		
Criterion	100	250	500	100	250	500
Raw confidence	0.40 (0.87)†	0.64 (0.92)	0.73 (0.93)	0.34 (0.94)	0.61 (0.95)	0.95 (0.96)
Platt	0.03 (0.20)†	0.79 (0.92)	0.85 (0.92)	0.07 (0.61)†	0.73 (0.68)†	1.00 (0.96)
Isotonic	—	**0.89 (0.90)**	0.85 (0.91)	0.06 (0.46)†	0.78 (0.70)†	1.00 (0.96)
KNN agreement	**0.64 (0.93)**	0.80 (0.92)	0.84 (0.92)	0.38 (0.94)	**0.89 (0.95)**	0.99 (0.97)
Platt+KNN	0.08 (0.79)†	0.74 (0.94)	0.76 (0.95)	0.10 (0.94)	0.39 (0.95)	0.95 (0.98)
Class-conditional	0.53 (0.91)	0.67 (0.93)	0.74 (0.94)	0.50 (0.94)	0.83 (0.96)	0.93 (0.97)
Outlier rejection	0.39 (0.87)†	0.62 (0.92)	0.70 (0.94)	0.33 (0.95)	0.58 (0.95)	0.93 (0.96)
Ensemble agreement	0.56 (0.94)	0.72 (0.93)	0.76 (0.95)	0.31 (0.94)	0.84 (0.94)	1.00 (0.96)
SVM margin*	0.63 (0.91)	0.80 (0.91)	**0.88 (0.91)**	**0.54 (0.91)**	0.87 (0.92)	**1.00 (0.97)**

with least-confidence uncertainty sampling [5], which selects tickets whose most probable class has the lowest probability. Both strategies label the same number of tickets, so differences in pseudo-labelling quality reflect which tickets were labelled rather than how many. Models are frozen at checkpoints of 100, 250 and 500 labelled tickets, and every criterion is applied to the remaining unlabelled pool. Since the process is simulated, the ground truth of every pool ticket is known and each criterion can thus be scored.

4.4 Metrics

At each checkpoint, let P denote the remaining unlabelled pool and $A \subseteq P$ the subset of tickets that a criterion labels, with $\hat{y}_i$ the label assigned to ticket i and y_i its true team.

Coverage is the fraction of the unlabelled pool that a criterion labels, and therefore the manual effort it saves:

$$\text{Coverage} = \frac{|A|}{|P|}. \tag{1}$$

Accuracy is the proportion of pseudo-labelled tickets with the correct label:

$$\text{Accuracy} = \frac{1}{|A|} \sum_{i \in A} \mathbb{1}[\hat{y}_i = y_i]. \tag{2}$$

For every criterion, we sweep its confidence parameter and report the result with the highest coverage subject to accuracy of at least 90%. Because the simulation knows the ground-truth labels of the remaining pool, this sweep is retrospective: each reported operating point is the best achievable configuration for that criterion at that checkpoint and forms an upper bound rather than a threshold that could be set in a live deployment, where the labels of the remaining tickets are unavailable. In such a deployment, thresholds would have to be fixed in advance, e.g. on held-out labelled data; the class-conditional criterion is the only one whose thresholds are estimated this way, and a validation-based selection for the remaining criteria is left to future work.

5 Results

Table 1 reports coverage and achieved accuracy at each criterion's configuration, that achieved the highest coverage subject to the accuracy of at least 90%. Results are shown for the three checkpoints and averaged over the three base classifiers; all operating points are retrospective upper bounds (Section 4). Daggered cells use the best-accuracy operating point when the 90% target cannot be met. A dash means that the criterion labelled no ticket.

At 100 labelled tickets, only some criteria reach the accuracy floor: k-nearest-neighbour agreement performs best under random sampling with 0.64 coverage at 0.93 accuracy, and the SVM margin leads under uncertainty sampling with 0.54 coverage at 0.91 accuracy. At 250 labels, k-nearest-neighbour agreement leads under uncertainty sampling with 0.89 coverage at 0.95 accuracy. At 500 labels, Platt and isotonic calibration, ensemble agreement and the SVM margin all reach full pool coverage at 0.96 to 0.97 accuracy under uncertainty sampling, while the combined criterion achieves the highest accuracy overall at 0.98 with 0.95 coverage.

The most dependable criterion overall is k-nearest-neighbour agreement. It is the only criterion that meets the 90% target at 100 labels under both strategies and still covers at least 80% of the pool at 250 and 500 labels. The composite criteria add complexity without improving results: outlier rejection consistently covers less of the pool than the raw confidence criterion it extends, and the combined criterion never improves on its k-nearest-neighbour parent.

5.1 Coverage and Accuracy

Figure 1 shows the Coverage-Accuracy trade-off over the whole sweep, complementing the single selected operating points of Table 1. The 100-label panel is sparse because most criteria contain no threshold reaching 90% accuracy, while the remaining criteria cover at most two-thirds of the pool. This is the only checkpoint where random sampling outperforms uncertainty sampling. From 250 labels onwards, the panels fill in and the dashed uncertainty curves consistently lie above the solid random-sampling curves at the same accuracy. Two effects may contribute: the model improves faster per acquired label under uncertainty sampling, and its queries focus on difficult tickets, leaving an increasingly easy unlabeled pool. Random sampling leaves more difficult tickets unlabelled. Pseudo-labeling and AL are therefore complementary: as the process queries more tickets, the model can label a larger share of the remainder itself. While the plot contains averaged results, the per-model frontiers can differ substantially.

5.2 Calibration at Low Label Budgets

While the calibration criteria are one of the best performing at 500 labels, reaching full coverage at 0.96 accuracy, they are not

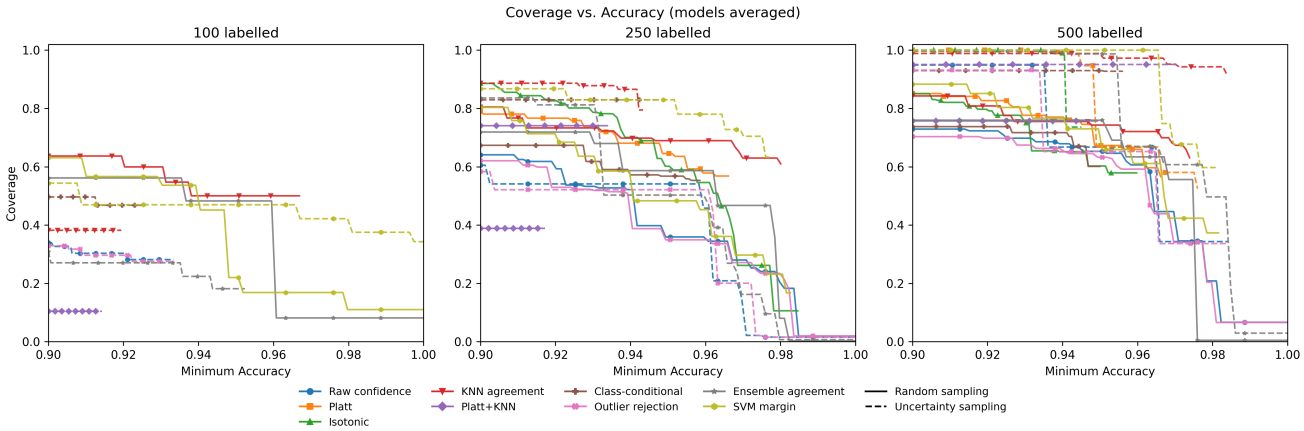


Figure 1: Coverage versus minimum accuracy for the nine criteria, one panel per checkpoint. Each curve averages the three base classifiers; solid lines show random sampling and dashed lines uncertainty sampling. At accuracy level x, the curve height is the highest coverage achievable subject to an accuracy of at least x; criteria whose accuracy never reaches x are absent from the panel.

the most robust. At the 100-label checkpoint, they collapse completely. The pseudo-labeled subset of the SVM is almost always wrong (accuracy close to zero), and logistic regression collapses similarly under random sampling. Isotonic regression fails differently. Under random sampling it auto-labels no ticket at all at 100 labels, and for logistic regression it collapses at 100 and 250 labels under uncertainty sampling. Both failures have the same root: the per-class recalibrations are estimated from a 235-ticket held-out split with 21 classes. Rare teams contribute only two or three positive examples, while a model trained on 100 tickets produces scores that are too weakly separated for a stable fit. Calibration therefore pays off only after the underlying model is accurate; agreement-based signals are safer while at the beginning of the process.

6 Conclusion

We studied nine pseudo-labeling criteria that decide which unlabelled tickets a model may label during an AL process, evaluated at checkpoints of 100, 250 and 500 labelled tickets on an IT support ticket dataset. Automatic labelling pays off once enough human labels have been collected: with 500 labelled tickets, the best-performing criteria label the entire remaining pool with 96 to 98% accuracy. Agreement-based signals remain dependable throughout, while probability calibration is unstable below roughly 250 labels and becomes competitive with the best agreement- and margin-based criteria once the underlying model is accurate.

6.1 Future Work

Two extensions follow naturally. First, the auto-labelled tickets are currently scored but not used for training. A next step is to retrain the model on the union of human and automatic labels and measure whether this accelerates the process or whether incorrect labels compound over iterations. The thresholds would also need to be adapted to prevent this. Second, all nine criteria are heuristics whose accuracy is observed but not bounded. Conformal prediction could replace them with criteria that provide a distribution-free guarantee on the proportion of wrong labels. Whether such guarantees survive the distribution shift introduced by AL remains to be seen.

Acknowledgements

This work was supported by the European Union's funded Project HUMAINE [grant number 101120218]. The dataset was provided by the project partner GFT, as a part of a pilot led by Michele Canteri.

References

[1] Shahan Ahmed, Anisur Rahman, and Md Ashrafuzzaman. 2023. A systematic review of AI and machine learning-driven IT support systems: enhancing efficiency and automation in technical service management. *American Journal of Scholarly Research and Innovation*, 2, 2, 75–101. doi:10.63125/fd34sr03.

[2] Qiang Fang and Xin Xu. 2026. Adapl: adaptive pseudo labeling for deep active learning in image classification. *Pattern Recognition Letters*, 199, 185–190. doi:10.1016/j.patrec.2025.11.024.

[3] Wenjing Han, Eduardo Coutinho, Huabin Ruan, Haifeng Li, Bjorn Schuller, Xiaojie Yu, and Xuan Zhu. 2016. Semi-supervised active learning for sound classification in hybrid learning environments. *PLoS ONE*, 11, 9, e0162075. doi:10.1371/journal.pone.0162075.

[4] Dong-Hyun Lee. 2013. Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*.

[5] David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM/Springer, 3–12.

[6] Burr Settles. 2009. Active Learning Literature Survey. Tech. rep. 1648. University of Wisconsin–Madison.

[7] Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. 2023. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112, 3211–3260. doi:10.1007/s10994-023-06336-7.

[8] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Han Zhang, Zizhao Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*. Vol. 33.

[9] Alaa Tharwat and Wolfram Schenck. 2023. A survey on active learning: state-of-the-art, practical challenges and research directions. *Mathematics*, 11, 4, 820. doi:10.3390/math11040820.

[10] Masahito Toba, Seiichi Uchida, and Hideaki Hayashi. 2024. Pseudo-label learning with calibrated confidence using an energy-based model. In *Proceedings of the International Joint Conference on Neural Networks*. doi:10.1109/IJCNN60899.2024.10650805.

[11] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*. Vol. 34.

[12] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17, 11, 1529–1541. doi:10.1109/TKDE.2005.186.

LLM-Generated Song Lessons: Perceived Usefulness and Shared-Cache Cost Analysis

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

LangSong is a mobile-first prototype that turns supplied song lyrics into a language lesson using a schema-constrained large language model (LLM), with analyses and exercises cached across users. We examine perceived feature usefulness, logged use and modelled cost per user. In a self-guided convenience study with 16 participants, tap-to-gloss and line translation received useful ratings from 11/16 and 10/16 respondents; transliteration, shown only for non-Latin scripts, was rated 3 to 5 by the four who could see it. Free exploration after the tour lasted a median of 0.8 minutes, and 11 ratings referred to features the participant had never opened. Per-call costs measured through the API (918 calls over three songs) feed a cost model: with the shared store, the model cost per user at 15 songs a month falls from $0.1406 at 100 users to $0.0211 at a million ($0.2034 without sharing), and lessons for the whole Italian catalogue of 2,500,000 songs are a one-off of about $30,625, a year of 851 subscribers. These are scenarios, not measured operating costs, and cover inference only; lyrics licensing and streaming-service terms were not assessed and may prove the harder constraints.

Keywords

computer-assisted language learning, large language models, song lyrics, perceived usefulness, caching, inference cost

1 Introduction

Songs help with learning a language. When listening to a song in the language being learned, each replay is an unplanned rehearsal [1], and repeated listening builds vocabulary without deliberate study [2, 3]. Following such a song is harder than it should be: a listener who wants to understand, for example, a Macedonian song switches between a streaming service, a lyrics site, a transliteration if they do not know the script, and a published translation if one exists or a translation tool such as a Large Language Model (LLM). Apps for learning languages through songs exist, but they have several limitations (Section 2).

We developed LangSong to cover all of this ground and more, as a novel approach that relies on what LLMs now make feasible. One schema-constrained LLM call turns any supplied lyrics into a lesson: transliteration, line by line translation, per-word glosses and grammar notes. From the lesson the learner can open a deeper explanation of the grammar of any word, ask an AI tutor questions, practise generated exercises, or just follow the real time lyrics while the song plays in the app. No lesson is written by hand, so any song can be covered, instead every lesson, Deeper card and exercise set comes from an LLM call. We evaluated the usefulness of this approach via user testing and survey, while also asking how the number of users affects the marginal cost of analysis. Output quality and rights to the source lyrics are separate questions, taken up in Section 6.

We ask two research questions. **RQ1** (value): which features do learners rate most useful? **RQ2** (cost): what is the marginal LLM cost of each feature, and how does a shared store of finished analyses change cost per user as the number of users grows?

2 Background and Related Work

2.1 Music, LLMs and language learning

Singing new phrases aids recall more than speaking them [2], and repeated listening yields incidental vocabulary gains [3]. Apps like LingoClip (formerly LyricsTraining) offer gap-fill, karaoke and a word translator over a selected catalogue, not lessons for any song in a streaming catalogue such as Spotify's [4, 5]. Duolingo's paid tier adds AI conversation practice [6] but no song-based course.

LLMs can explain word meanings in context, produce grammar notes on demand and generate exercises [7, 8]; generated exercises have been rated close to human-written ones [9]. Generative-AI research in language learning is concentrated on learners of English and on chatbots [10]; song-based, multi-language use is largely unexplored.

2.2 Popularity skew

LangSong stores each finished analysis and serves it to every later listener, so whether that saves cost depends on how concentrated demand is: web requests follow Zipf-like distributions, where the i-th most popular item is requested about $1/i$ as often as the first, and cache performance depends on this skew and on catalogue size [11]. Music demand is similarly skewed [12, 13], and the long tail is where "any song" matters [14].

3 The LangSong Prototype

3.1 Features

LangSong is built as a web app requiring three main components: a streaming service (Spotify), lyric data (Musixmatch) and an LLM provider (OpenAI). The song is in the language being learned; the language the app translates and explains into, which we call the target language, is currently English only. It syncs with the user's streaming library, and with one click any of their songs can be translated. For each song one LLM call returns a lesson: line-by-line transliteration in a per-language Latin standard, line-by-line translation, for every word its dictionary form, part of speech and short meaning (a gloss), and four to six grammar notes. Once in the lesson, the user can listen as in any streaming service while the current line is highlighted with its transliteration and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.111

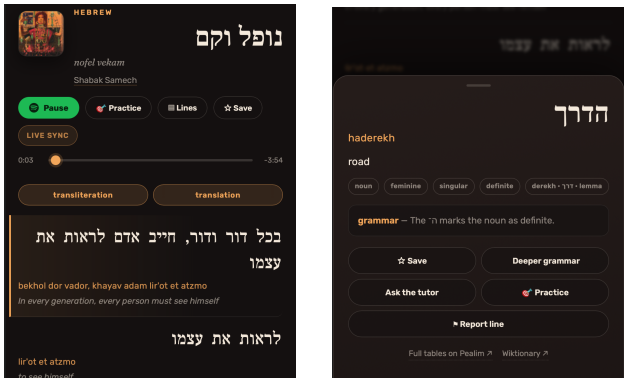


Figure 1: The lesson view during playback, with the current line highlighted and its transliteration and translation (left), and a word card with the gloss, grammar tags and the Deeper, tutor and practice buttons (right).

translation (Figure 1; timed lyrics were available for 87.8% of the songs with lyrics in the prototype's catalogue). Two song-level exercises follow: the Lines trainer plays one line at a time and asks the learner to fill or spell a missing word, and Practice offers ten generated exercises per song: word meanings in both directions, a lyric line with a blank, questions on the grammar notes, the vocabulary in new sentences, and two typed answers in the original script. Tapping any word opens its card: dictionary form, part of speech (noun, verb, …), meaning and a one-line grammar note, with buttons to save the word, open a Deeper grammar explanation (the word's full morphology, forms and examples), ask the AI tutor (a chat that knows the word and its line in the song) or practise the word (exercises generated for that word).

3.2 One-call analysis and the shared cache

Lyrics are normalised and hashed, and the hash names the cached analysis, shared by all users; two catalogue entries with the same lyrics (a remaster, a re-recording) are therefore analysed once. Only 63.3% of the lines in our corpus are unique (4,141 of 6,540); the model annotates each unique line once and the app copies the result to its repeats, to save output tokens (the full lyrics still go in as context). The call uses OpenAI `gpt-5.6-luna` through the Codex CLI at low reasoning effort with JSON-schema output. A validator checks that every line came back with a Latin-only transliteration (if lyrics are in non-Latin script) and has correct structure; a follow-up call repairs lines if there are errors. Deeper cards and practice sets are stored by word and song and shared the same way; the song level practice sets are pre-generated right after analysis, so their cost is shared too. AI Tutor sessions are not cached.

The app runs on a single small virtual server. The snapshot used below holds the songs analysed so far by the connected users and the study: 148 songs in 14 languages, Hebrew the largest group, with 59 Deeper grammar cards.

4 User Study Results

4.1 Protocol and observation window

The self-guided study involved 16 participants on their own phones (age bands 18–24 through 55+; 14 native South Slavic speakers; 12 past Duolingo users). Participants received a personal link, chose a language and song, followed an in-app tour

Table 1: Optional features: used at any point in the study (one participant also on an earlier account) and rated useful (4 or 5) by those who rated it.

Feature	Used	Rated useful
Deeper grammar card	13	5/10
Tutor chat	6	1/5
Practice exercises	12	5/11

of the features, then explored before completing a questionnaire. They tested 7 languages; asked for their knowledge level in that language, 6 answered none or a few words, 6 A1–A2 and 4 B1–B2. The questionnaire asked participants to rate how useful each feature was for understanding the song (perceived usefulness [15]), from 1 to 5 or "didn't try"; to name the one feature they would keep and the one they trusted least; and to say what they would use LangSong for, in which form (standalone app, inside Spotify, browser extension), what they would pay per month and whether the translations looked accurate. All 18 exported responses include consent and distinct participant codes; logs were joined by normalised code. Two respondents whose native language was the language they tested are excluded, leaving 16. Two participants left the tour for the practice exercises before its last steps; for one the in-app browser also reloaded the page mid-tour.

Logged use is reported alongside the ratings, because stated and revealed preference can diverge [16]. After the tour, the 14 participants with a completed tour explored for a median of 0.8 minutes (mean 1.8) before opening the questionnaire.

4.2 Ratings

Figure 2 shows the ratings. A rating counts only when the log shows the participant used the feature, in the study or on an earlier account: 11 answers rated a feature the participant never opened and are treated as "didn't try". Tap-to-gloss was rated 4 or 5 by 11/16 respondents, followed by translation (10/16), practice (5/11), Deeper (5/10) and tutor chat (1/5). Transliteration appears only for non-Latin scripts: the twelve Latin-script testers rated it anyway, those ratings are dropped, and the four Russian and Japanese testers rated it 5, 5, 4 and 3. Asked which single feature they would keep, 6 chose translation, 4 gloss, 3 practice, 2 Deeper, 1 transliteration and 0 the tutor; asked which they trusted least, most named the tutor (5; 6 named none).

4.3 Logged use

The tour makes every participant tap a word and lets them try or skip the other steps; Table 1 counts who used each optional feature at any point in the study. The tutor was used by 6 participants, of whom 1/5 rated it useful. Free exploration after the tour was brief (median 0.8 minutes), and not every rating reflects use, as the transliteration answers show; Figure 2 therefore shows the data after cleaning.

5 Cost and Scale

5.1 Measurement

To emulate what a production deployment would use, we processed 3 cached songs (Hebrew, Russian and German) through the OpenRouter API with `gpt-5.6-luna` and DeepSeek V4 Flash: the lesson (analysis, a repair when the validator asks for one, which happened once, on Luna; the practice set), then a Deeper card and a word-practice set for every unique word, and one tutor turn,

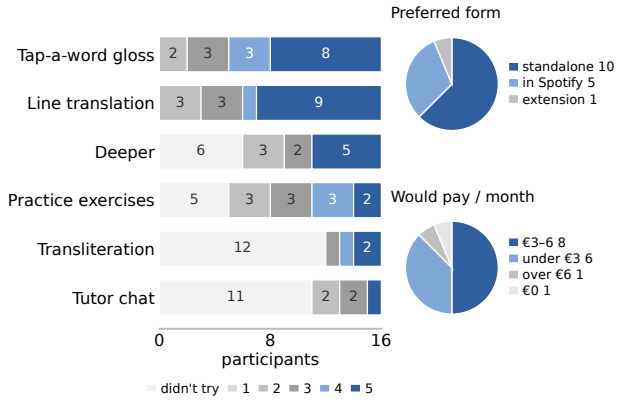


Figure 2: Questionnaire results after cleaning ($n = 16$): native-language testers excluded, transliteration counted only for non-Latin-script testers, features never opened counted as not tried.

Table 2: Measured API cost per call with Luna: means over 3 songs, n calls, input and output tokens, the share of input served from the prompt cache, and billed USD.

Feature	n	Input	Output	Cached	USD
Song analysis	3	2,029	7,904	0%	0.0100
Song practice	3	4,736	895	0%	0.0023
Deeper card	222	1,437	649	90%	0.0008
Word practice	222	1,190	600	59%	0.0008
Chat turn	3	599	85	0%	0.0002

919 billed calls in all ($0.52); one DeepSeek Deeper call failed validation ($0.02) and is left out of Table 2, which reports the cost per call. The prompts are the app's own, reordered so that the fixed rules and the lyrics form a shared prefix for provider prompt caching (the cached share in Table 2). OpenAI caches only prefixes of at least 1,024 tokens, so the calls made once per song, which share only their short instructions, show no cached input, while the word calls share the rules and the lyrics. DeepSeek ran with thinking off (with it on, one analysis cost as much as Luna's).

Costs are the billed amounts at the providers' rates [17]: Luna $0.20 per million input tokens, 1.25 times that for a prompt of at least 1,024 tokens being written to the cache, $0.02 when read from it and $1.20 per million output tokens; V4 Flash as hosted on OpenRouter $0.08, $0.016 and $0.16. Optional tools, fast processing and regional uplifts are outside the scenario.

5.2 Model

Let π_{in}, π_c and π_{out} be the prices per input, cached-input and output token, and t_{in}, t_c, t_{out} the input, cached and output tokens of one call of feature f (Table 2 uses the billed values directly):

$$C_f = \pi_{\text{in}}(t_{\text{in}} - t_c) + \pi_c t_c + \pi_{\text{out}} t_{\text{out}}. \tag{1}$$

Let songs be requested with Zipf probabilities $p_i = i^{-s}/H_{M,s}$ (the i-th most popular song is requested in proportion to i^{-s}), $H_{M,s} = \sum_{i=1}^{M} i^{-s}$, over a catalogue of M songs and each user make k independent song requests per month, with replacement; after N users the expected number of distinct songs analysed is

$$\mathbb{E}[D(N)] = \sum_{i=1}^{M} \left(1 - (1 - p_i)^{Nk}\right). \tag{2}$$

A new song costs at least its lesson, $C_{\text{analysis}} + C_{\text{prac}}$. Opening a word costs nothing, since its card comes from the analysis; asking for a word's Deeper card costs C_{deeper} and practising the word a word-practice set, C_{wordprac}. The two are requested independently, so the cost of one new song grows with the Deeper cards w_d and word-practice sets w_p generated for it, from the lesson alone ($w_d = w_p = 0$) to every unique word (Table 3):

$$C_{\text{new}} = C_{\text{analysis}} + C_{\text{prac}} + w_d\, C_{\text{deeper}} + w_p\, C_{\text{wordprac}}. \tag{3}$$

The baselines $w_d = 0.97$ and $w_p = 0.53$ are the Deeper cards and word-practice sets per song in production (33 Deeper cards and 18 sets over the 34 songs on which any word was opened). Repair calls (35 after 21 of the 83 logged analyses, some needing several rounds) cost $0.0015 each on Luna and are left out of C_{new}: at the logged rate they add under 5% to the bundle and can likely be eliminated with improvements to the system. With F the fixed monthly infrastructure cost and q AI tutor chat turns per user-month, the monthly cost per user with the shared cache, the same without any cache, and the cache hit rate are

$$c(N) = \frac{C_{\text{new}}\,\mathbb{E}[D(N)]}{N} + q\,C_{\text{chat}} + \frac{F}{N}, \tag{4}$$

$$c_{\text{nc}}(N) = k\,C_{\text{new}} + q\,C_{\text{chat}} + \frac{F}{N}, \tag{5}$$

$$h(N) = 1 - \mathbb{E}[D(N)]/(Nk). \tag{6}$$

The scenario sets $k = 15$ song requests per user-month, $q = 3$ chat turns, $s = 1$, $F = 0$ so that the curve is the model cost alone, and $M = 2,500,000$, an Italian-sized catalogue: Italian is 2–3% of the WASABI lyrics corpus [18, 19] and Spotify holds over 100 million tracks [20], while MusicBrainz lists 44,449 Italian-language and 112,702 Italian releases, on the order of 300,000 to 800,000 recordings [21]. τ = $0.10 per user-month, a thirtieth of the subscription below, is a reference point for comparing scenarios, not a target. Equation 2 models one month from an empty store; a persistent catalogue costs less in later months.

5.3 Scale

Without sharing, a user who requests 15 songs a month costs $0.2034 in model calls. With the shared store the cost per user goes from $0.1406 per month at 100 users to $0.0797 at 10,000 and $0.0211 at 1,000,000, as the hit rate rises from 31.0% to 89.9%, towards the AI tutor cost of $0.0007. Ten cents per user-month is reached at 2,152 users; the fall is slow because at 15 requests a month a large catalogue keeps being explored. With DeepSeek V4 Flash's measured costs the cost per user is $0.0334 at 100 users and $0.0050 at a million, under ten cents even without sharing ($0.0483).

Against a hypothetical subscription of $3 a month, a user costs more than they pay only through generation nobody has requested before: at $0.0148 per new song with 1.5 Deeper cards and as many word-practice sets, a user must add 203 new songs a month before their model cost passes the subscription, or 22 if every word of every song is generated (853 and 95 with Flash). A cap of 50 new analyses per user-month keeps new-song generation at that rate to $0.74 (25% of the subscription); word-level generation on songs already analysed and chat would need limits of their own. Such a cap is worth enforcing for niche languages, where few later listeners share the analysis; for popular pairs

Table 3: Generating ahead of demand: cost per song as a lesson only (analysis and practice) and fully pre-generated (plus a Deeper card and a practice set for each of the median 74.5 unique words); the one-off cost of an Italian-sized catalogue of 2,500,000 songs either way; and the number of subscribers at a hypothetical $3 a month whose first year covers it.

Model	Per song		Whole catalogue		Subscriber-years	
	Lesson	Full	Lessons	Full	Lessons	Full
Luna	$0.012	$0.14	$30,625	$345,204	851	9,589
DeepSeek V4 Flash	$0.003	$0.03	$7,352	$79,293	205	2,203

such as English speakers learning Italian, every new analysis populates the catalogue for other users, so heavy users are cheap content acquisition. Table 3 turns the same per-call costs into what generating a whole catalogue ahead of demand would cost.

6 Discussion and Limitations

Feature interpretation. Useful ratings for Deeper, practice and the tutor came after a short session that the tour had just walked through, and some were given without use. The result supports a simple annotated-song screen with optional learning controls; it does not show that learners reject those features.

What participants want. Respondents most often preferred a standalone app (10/16), followed by an in-Spotify feature (5) and a browser extension (1). Spotify and YouTube Music each received 11 playback-integration selections. AI-assisted playlists received 10 votes and lesson packs 9. 15/16 selected a nonzero hypothetical monthly price (not observed purchasing): 6 under €3, 8 at €3–6 and 1 above €6. Learning and song understanding received 12 and 12 selections, with 8 choosing both. Of 14 respondents able to judge perceived translation accuracy, 11 answered yes and 3 mostly; there was no independent linguistic evaluation.

Where scale helps. On demand, the modelled inference cost per user stays below fifteen cents a month even at a hundred users ($0.1406), so on modelled inference cost alone a hypothetical $3 subscription covers a language offered on demand from the start; the question is when generating ahead of demand pays off. Lessons for the whole Italian catalogue of 2,500,000 songs are a one-off of about $30,625 with Luna or $7,352 with DeepSeek V4 Flash (Table 3), the first year of 851 or 205 subscribers at $3 a month; scenario catalogues of 300,000 songs (about the Hebrew repertoire) and 100,000 (Macedonian) cost $3,675 and $1,225. Pre-generating a Deeper card and a practice set for each of the 74.5 unique words of every song raises the Italian figure to $345,204 (9,589 subscriber-years), worth doing only for the head of the catalogue. Each target language is a separate lesson (translation and glosses change), so language pairs multiply the catalogue cost; English-target pairs cover most learners, and niche pairs such as Hebrew songs for Macedonian speakers stay on demand.

Cost and limitations. Shared caching can reduce modelled inference cost, but the result depends on request concentration, catalogue size, word practice, chat demand and per-call costs measured on three songs through one API gateway, whose cached shares come from bulk generation with a warm prefix that sparse on-demand word calls would not reach. It excludes licensing, support, acquisition and infrastructure scaling. The cost analysis rests on one usage model, the same for every user: k independent requests a month from one Zipf catalogue at the production rates of Deeper cards and word practice. Real use could differ substantially; whether users keep to one language pair or several, and outlier users, are out of scope. Under these assumptions inference cost appears manageable for a wider deployment; lyrics licensing and streaming-service terms may matter more, but neither was assessed (no licensing quote or feasibility study). The study is a small convenience sample recruited from the author's network, using shared playback and self-reported proficiency. Device and playback failures can depress logged use; the counts support no causal or population-wide conclusions.

7 Conclusion

LangSong demonstrates shared generation and reuse of song-based language annotations. In a 16-person formative study, gloss and translation received the most useful ratings; free exploration after the tour was brief, and some ratings referred to features the participant had not opened. Cost per user falls with the number of users because every analysis is shared, from $0.1406 to $0.0211 per month between a hundred and a million users under the baseline assumptions, and lessons for a whole Italian-sized catalogue cost about a year of 851 subscribers. Under those assumptions inference cost appears manageable; lyrics licensing and streaming-service terms, not assessed here, may prove the harder constraints. Further work should assess them alongside usage, learning and capacity, and evaluate the lessons linguistically.

Acknowledgements

The research was supported through the European project Raido (GA 101135800). The author acknowledges usage of LLMs in grammar refinement and overall manuscript organisation and takes full responsibility for the final content of this publication.

References

[1] Tim Murphey. 1990. The song stuck in my head phenomenon: a melodic din in the LAD? *System*, 18, 1, 53–64. doi:10.1016/0346-251X(90)90028-4.

[2] Karen M. Ludke, Fernanda Ferreira, and Katie Overy. 2014. Singing can facilitate foreign language learning. *Memory & Cognition*, 42, 1, 41–52. doi:10.3758/s13421-013-0342-5.

[3] Niousha Pavia, Stuart Webb, and Farahnaz Faez. 2019. Incidental vocabulary learning through listening to songs. *Studies in Second Language Acquisition*, 41, 4, 745–768. doi:10.1017/S0272263119000020.

[4] LingoClip. 2025. LingoClip (formerly LyricsTraining): learn languages with music. https://lingoclip.com/. Accessed 2026-08-28; previously https://lyricstraining.com/. (2025).

[5] Brian S. McKinney. 2025. LyricsTraining and LingoClip. *CALICO Journal*, 42, 2, 336–344. doi:10.3138/calico-2024-0005.

[6] Duolingo. 2023. Introducing Duolingo Max, a learning experience powered by GPT-4. https://blog.duolingo.com/duolingo-max/. (Mar. 2023).

[7] Lucas Kohnke, Benjamin Luke Moorhouse, and Di Zou. 2023. ChatGPT for language teaching and learning. *RELC Journal*, 54, 2, 537–550. doi:10.1177/00336882231162868.

[8] Enkelejda Kasneci et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. doi:10.1016/j.lindif.2023.102274.

[9] Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating reading comprehension exercises generated by LLMs: a showcase of ChatGPT in education applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. Association for Computational Linguistics, Toronto, Canada, 610–625. doi:10.18653/v1/2023.bea-1.52.

[10] Belle Li, Yaling Lily Tan, Chaoran Wang, and Victoria Lowell. 2025. Two years of innovation: a systematic review of empirical generative AI research in language learning and teaching. *Computers and Education: Artificial Intelligence*, 9, 100445. doi:10.1016/j.caeai.2025.100445.

[11] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. 1999. Web caching and Zipf-like distributions: evidence and implications. In *Proceedings of IEEE INFOCOM '99*. Vol. 1. IEEE, New York, NY, 126–134. doi:10.1109/INFCOM.1999.749260.

[12] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311, 5762, 854–856. doi:10.1126/science.1121066.

[13] Peter L. Ormosi and Franco Mariuzzo. 2022. Independent vs major record labels: do they have the same streaming power (law)? *Economics Bulletin*, 42, 3, 1510–1526. https://www.accessecon.com/Pubs/EB/2022/Volume42/EB-22-V42-I3-P126.pdf.

[14] Chris Anderson. 2006. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, New York. ISBN: 978-1-4013-0237-5.

[15] Fred D. Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13, 3, 319–340. doi:10.2307/249008.

[16] Paul A. Samuelson. 1938. A note on the pure theory of consumer's behaviour. *Economica*, 5, 17, 61–71. doi:10.2307/2548836.

[17] OpenAI. 2026. API pricing: standard short-context rates. https://developers.openai.com/api/docs/pricing. Rates recorded 2026-08-28. (2026).

[18] Michael Fell, Elena Cabrio, Elmahdi Korfed, Michel Buffa, and Fabien Gandon. 2020. Love me, love me, say (and write!) that you love me: enriching the WASABI song corpus with lyrics annotations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC 2020)*. European Language Resources Association, Marseille, France, 2138–2147. https://aclanthology.org/2020.lrec-1.262/.

[19] Michel Buffa et al. 2021. The WASABI dataset: cultural, lyrics and audio analysis metadata about 2 million popular commercially released songs. In *The Semantic Web (ESWC 2021)* (Lecture Notes in Computer Science). Vol. 12731. Springer, 515–531. doi:10.1007/978-3-030-77385-4_31.

[20] Spotify. 2026. About Spotify. https://newsroom.spotify.com/company-info/. Over 100 million tracks. Accessed 2026-08-31. (2026).

[21] MusicBrainz. 2026. Database statistics: languages and scripts; countries. https://musicbrainz.org/statistics/languages-scripts. Accessed 2026-08-31. (2026).

Reliable but Not Right: Automated Evaluation of Generated R&D Ideas

Adrian Mladenić Grobelnik
adrian.m.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

Businesses regularly face the decision of which research ideas to fund, and an automated ranking of the candidate research and development ideas can provide decision support for it. As with any business intelligence tool, the ranking is worth acting on only if it reflects the judgment of the decision maker. Automated research systems generate more candidates than anyone can read, and large language models are commonly used to rank them. Such an automated evaluator is validated along several dimensions, accuracy on past work, self-consistency, position bias and agreement between models, none of which needs a human expert to look at the candidates it will rank. We propose a design for validating automated evaluators of generated research and development ideas against a human expert. We applied it in three studies: a citation-impact predictor choosing the starting concepts (seeds) for hypotheses, large language models ranking generated hypotheses, and language models ranking generated papers. We were surprised that in each study the evaluator passed its automated validation on every dimension, yet its ranking diverged from the human expert's: it was reliable but not right. Our experimental validation showed that recognising a good research idea involves something experts do that current models do not.

Keywords

LLM-as-judge, automated evaluation, human evaluation, research idea generation, self-consistency, position bias, citation impact prediction, research and development, decision support

1 Introduction

Research and development (R&D) runs on a choice no one can make well at scale: which ideas to fund before their outcome is known. Decision makers can read only a fraction of what is proposed. Automated research systems widen this gap because they generate hypotheses, experiments and papers faster than any reviewer can read them [1, 2]. A large language model is therefore asked to rank the candidates before a person sees them [3]. That ranking is a business intelligence tool [4] for the funding decision, but it helps only when the automated evaluator ranks the candidates the way the decision maker would have.

This paper asks whether an evaluator that passes validation without a human expert also ranks candidates the way a human expert would. We tested the question in three studies: (1) a citation-impact predictor, a model trained on past papers with known outcomes, chose the starting concepts for generating hypotheses; (2) large language model judges ranked generated hypotheses; and (3) large language model judges ranked generated papers. Each evaluator was validated along several dimensions before it ranked anything. By concept we mean a Wikipedia entry that a paper's title or abstract mentions.

The impact predictor was validated by its accuracy on held-out papers. The judges were validated on self-consistency, whether they return the same verdict when a pair of candidates is shown again. They were also validated on position bias, whether the verdict changes when the two candidates swap places [5], and on agreement between judges. None of these requires a human expert to examine the candidates being ranked.

We propose a design for validating an automated evaluator of generated R&D ideas against a human expert, and apply it in all three studies.

Contributions.

(1) *A design for validating automated evaluators of generated R&D ideas.*
(2) *Insights into the ability of current models to recognise high-relevance research.*

2 Related Work

Large language model judges. Earlier work validates a large language model judge against human labels for the task it judges, but the judges in our studies had no such labels when they were chosen. Zheng et al. [3] found a large language model agreeing with human preference on more than 80% of chatbot conversation pairs, yet found position, verbosity and self-enhancement biases alongside that agreement. Wang et al. [5] showed that swapping two candidates can reverse the verdict. Chen et al. [6] found judges misled by authority cues and presentation. Bavaresco et al. [7] found that agreement between judges and human annotators varies widely across 20 tasks.

Automated research and its evaluation. Automated research systems now incorporate such judges. The AI Scientist [1], the pipeline of Mladenić Grobelnik [8], Co-Scientist [2] and SciMON [9] generate hypotheses or whole papers. All four use a large language model to review them, rank them or assess their novelty. Si et al. [10] had more than 100 NLP researchers review generated and human-written research ideas: the generated ideas were judged more novel but slightly less feasible. That is the kind of human evaluation an automated evaluator stands in for.

Impact prediction. Citation-impact predictors, the first kind of evaluator we study, have been validated against what was later published or cited rather than against an expert's view of the ideas themselves. Uzzi et al. [11] link citation impact to unusual combinations of prior work. Gu and Krenn [12] forecast

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.82

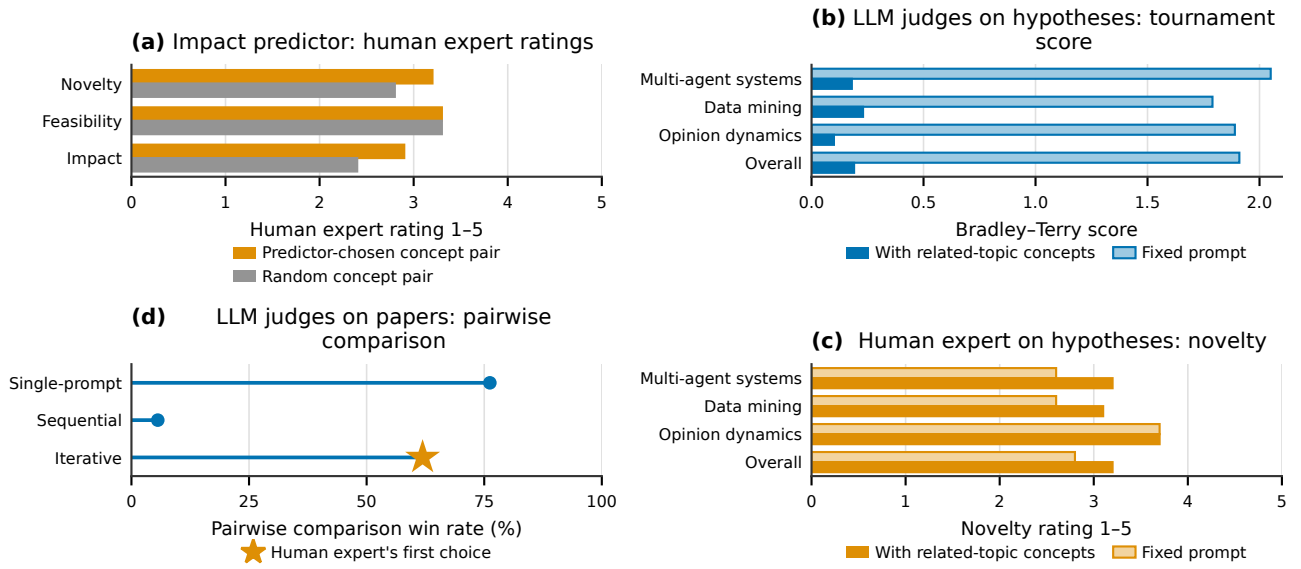


Figure 1: (a) The human expert's mean ratings of ten predictor-chosen and ten random hypotheses on novelty, feasibility and impact.(b) and (c) The 40 hypotheses by field: the judges preferred those without concepts from a related topic everywhere, the human expert those with them. (d) The three papers on the first hypothesis: the share of comparisons each strategy won when the judges saw only the paper. The star marks the human expert's first choice. Blue marks the automated evaluator's ranking, orange the human expert's.

the impact of concept pairs not yet published from an evolving knowledge graph. Vital et al. [13] predict citation impact from text embeddings. Outside research, business uses of large language models such as data preparation in analytics [14] and business process modelling [15] can be checked against a known answer.

3 The Proposed Design

Figure 2 shows the design: the automated evaluator first passes its validation without a human expert, and a human expert then assesses the candidates it ranked. Each study applies this design with a single expert: a domain expert, a domain researcher and a code reviewer respectively, each called the expert below.

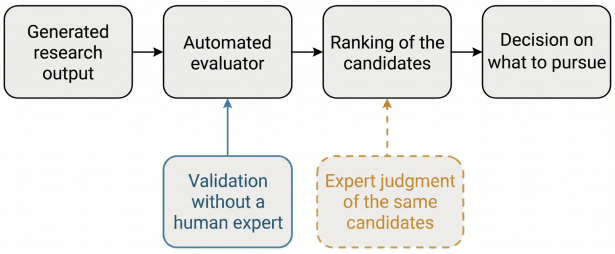


Figure 2: The proposed design for validating an automated evaluator of generated research output. The evaluator first passes its validation without a human expert, then a human expert assesses the candidates it ranked.

3.1 Citation-Impact Predictor for Generating Hypotheses

Can an accurate citation-impact predictor generate hypotheses that match a human expert's preferences? The idea is straightforward: train a model on papers with known citation outcomes, then let its predictions choose the starting concepts for generating hypotheses. We collected 1.71M distinct OpenAlex [16] papers from 2016 to 2026, the 200 highest-impact and 200 random papers in each of 4,516 topics. We trained nine models to predict field-weighted citation impact, a paper's citations divided by the average for its field and year. The inputs were an embedding of the title and abstract, metadata such as year and author count, and the venue's average impact. Further inputs were references shared with other papers, keywords and the concepts in the abstract. The impact predictor passed its validation without a human expert because it predicted the impact of held-out papers well. The strongest model, an AutoGluon ensemble [17], reached Spearman 0.73 with the true field-weighted citation impact on a held-out 25% over three runs. Its ROC-AUC at identifying the 10% most cited papers was 0.87. An FT-Transformer [18] followed at 0.72 and 0.86.

A domain expert then chose ten research directions in the field of multi-agent systems. For each direction a large language model generated two hypotheses, each from a concept pair. In the first, a concept derived from the research direction was paired with the highest-scoring concept, according to the impact predictor, among those not yet combined with it in any paper. In the second, the partner concept was chosen at random from those no paper had yet combined with the first. This gave a predictor-chosen set and a random set of ten hypotheses each, and the expert rated every hypothesis from 1 to 5 on novelty, feasibility and impact.

Result. The expert's ratings show no significant difference between the predictor-chosen set and the random set on novelty, feasibility or impact, as Figure 1a shows. The predictor-chosen set scored 3.2 against 2.8 on novelty, 3.3 against 3.3 on feasibility and 2.9 against 2.4 on impact, $p = 0.32$, 0.93 and 0.12. Ten ratings per set give little statistical power, so this is a null result, not evidence that the impact predictor chooses worse than random.

3.2 Large Language Model Judges on Generated Hypotheses

Can large language model judges that are self-consistent and positionally unbiased rank research hypotheses as a human expert would? Three large language models, none of which later served as a judge, generated 40 hypotheses across the domains of multi-agent systems, data mining and opinion dynamics from the same OpenAlex data [16]. Half the prompts named two concepts from a related topic, meaning concepts that papers in that topic use but papers in the target field do not. The other half used a fixed prompt without concepts from a related topic, giving 20 hypotheses with such concepts and 20 without.

We screened eight large language models as candidate judges on self-consistency and position bias, without a human expert. Table 1 reports the results: we kept the two models with the highest self-consistency, Claude Sonnet 4.5 [19] and GPT-5.2 [20], as the judges we compared with the expert. We expected their ranking to align with the expert's judgement.

Table 1: Validation of the eight candidate judges, without a human expert. Self-consistency is the share of pairs that return the same winner in both orders, higher is better. Position bias is how far the first-position win rate deviates from 50%, lower is better.

Model	Self-consistency	Position bias
Claude Sonnet 4.5	**100%**	**0%**
GPT-5.2	**89%**	**14%**
GPT-5-mini	80%	20%
GPT-5.2 Pro	79%	11%
Claude Opus 4.5	75%	25%
Gemini 3 Flash	75%	25%
gpt-oss-120b	75%	25%
Grok 4.1 Fast	65%	35%

The judges ranked all 40 hypotheses by pairwise comparison using the Bradley–Terry model [21], discarding any vote that reversed when the two candidates swapped places. The expert, a domain researcher, then independently rated every hypothesis from 1 to 5 on novelty and on feasibility.

Result. The judges preferred hypotheses without concepts from a related topic in every field, with a mean novelty Bradley–Terry score of 1.91 against 0.19. The expert rated hypotheses with concepts from a related topic more novel in multi-agent systems and data mining, equally novel in opinion dynamics, and 3.2 against 2.8 overall, $p = 0.055$, as Figure 1b and c show. Those same hypotheses were slightly less feasible, 3.5 against 3.8, $p = 0.16$. The judges and the expert agreed on feasibility, but on novelty the judges preferred hypotheses without concepts from a related topic while the expert preferred those with them. Within each field, the judges' novelty score and the expert's novelty rating had a Spearman correlation of 0.00 in multi-agent systems, 0.05 in data mining and 0.83 in opinion dynamics, over 17, 17 and six hypotheses.

3.3 Large Language Model Judges on Generated Papers

Can large language model judges that are self-consistent, positionally unbiased and in agreement with each other rank papers as a human expert would? Using the agentic pipeline of Mladenić Grobelnik [8], we compared three strategies for testing a hypothesis and writing it up as data, code and a paper, shown in Figure 3. The single-prompt strategy gives one agent the whole task. Fixed sequential execution splits it across three agents running one stage each, with no feedback. Iterative planning has agents propose strategies, pick the best, execute it and feed the results and what is still missing into the next round. Each strategy ran on two machine learning hypotheses, giving six runs.

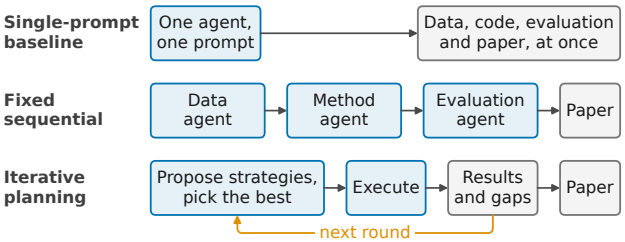


Figure 3: The three strategies for testing a hypothesis and writing it up as data, code and a paper. Only iterative planning feeds results back into the next round.

We screened candidate judges on self-consistency, position bias and cross-model agreement, without a human expert. Claude Sonnet 4.5 and GPT-5.2 passed and were used to rank the six runs by pairwise comparison [3], seeing only the paper, only the code, or both. The two judges agreed with each other on 89% of comparisons. A human expert independently assessed all six runs.[1]

Result. Figure 1d shows the pairwise comparison win rates of the judges. The judges ranked the three strategies six times: for each of the two hypotheses, once seeing only the paper, once only the code and once both. The expert put the iterative strategy first on both hypotheses, and the judges agreed with the expert in five of the six rankings. The exception was the first hypothesis judged on the paper alone, where the judges ranked the single-prompt strategy first. There the single-prompt strategy's paper claimed the hypothesis was confirmed by two small-scale tests. The expert found the claim unsupported by its code and preferred the iterative strategy, which ran 30 orderings and returned a negative finding. When the judges also saw the code they too preferred the iterative strategy.

4 Discussion and Limitations

Across the three studies, validation without a human expert did not replace the expert. Each automated evaluator passed its validation that required no human expert. Accuracy on past work measures whether the impact predictor has learned the patterns of published work. Self-consistency, position bias and agreement between judges measure whether the judges are stable. These checks are necessary, but not sufficient: only one of eight candidate judges returned the same verdict on every pair it saw twice. In each of the three studies, passing automated validation did not guarantee agreement with the human expert on generated R&D ideas.

[1]The six generated papers, their code and their outputs are available at tinyurl.com/24zx5bu7.

Limitations. There are several limitations to this work. Each study uses a single expert and a small sample: ten research directions, 40 hypotheses and six runs. The disagreement in each study could be specific to its expert, and ruling this out takes several experts per study, which is our next step. All three studies come from one pipeline and one research group. The claim is narrow: in three cases, passing validation without a human expert did not guarantee agreement with the expert.

Conclusion. We validated three automated evaluators of generated R&D ideas against a human expert: a citation-impact predictor on research directions, large language model judges on hypotheses, and language model judges on papers. Each passed its automated validation on every dimension, yet each diverged from the expert on part of what it ranked. The paper judges matched the expert on both hypotheses only once they saw the code. Recognising a good research idea involves something human experts do that current models do not. An organisation using such an evaluator to rank its R&D ideas should measure its agreement with a human expert before acting on it [22].

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency (ARIS) via the Gravity project AI for Science (GC-0001) and European HE projects DataPact (GA 101189771) and RAIDO (GA 101135800). The authors acknowledge the use of large language models to support paper organisation, drafting and figure preparation, and language and grammar refinement. All suggestions from the models were carefully reviewed and revised by the authors, who take full responsibility for the content of this paper.

References

[1] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: towards fully automated open-ended scientific discovery. arXiv: 2408.06292 [cs.AI].

[2] Juraj Gottweis et al. 2026. Accelerating scientific discovery with Co-Scientist. *Nature*, 655, 8122, 487–496. DOI: 10.1038/s41586-026-10644-y.

[3] Lianmin Zheng et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 46595–46623.

[4] Rimvydas Skyrius. 2021. *Business Intelligence: A Comprehensive Approach to Information Needs, Technologies and Culture. Progress in IS*. Springer. DOI: 10.1007/978-3-030-67032-0.

[5] Peiyi Wang et al. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9440–9450. DOI: 10.18653/v1/2024.acl-long.511.

[6] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? A study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 8301–8327. DOI: 10.18653/v1/2024.emnlp-main.474.

[7] Anna Bavaresco et al. 2025. LLMs instead of human judges? A large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 238–255. DOI: 10.18653/v1/2025.acl-short.20.

[8] Adrian Mladenić Grobelnik. 2025. *Automating the Research Process with Agentic Large Language Models*. Bachelor's thesis. University of Ljubljana, Faculty of Computer and Information Science. Retrieved Sept. 5, 2026 from https://hdl.handle.net/20.500.12556/RUL-174624.

[9] Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 279–299. DOI: 10.18653/v1/2024.acl-long.18.

[10] Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs generate novel research ideas? A large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, 94003–94092.

[11] Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. 2013. Atypical combinations and scientific impact. *Science*, 342, 6157, 468–472. DOI: 10.1126/science.1240474.

[12] Xuemei Gu and Mario Krenn. 2025. Forecasting high-impact research topics via machine learning on evolving knowledge graphs. *Machine Learning: Science and Technology*, 6, 2, 025041. DOI: 10.1088/2632-2153/add6ef.

[13] Adilson Vital, Filipi N. Silva, Osvaldo N. Oliveira, and Diego R. Amancio. 2025. Predicting citation impact of research papers using GPT and other text embeddings. *Physica A: Statistical Mechanics and its Applications*, 674, 130789. DOI: 10.1016/j.physa.2025.130789.

[14] Mehran Nasseri, Patrick Brandtner, Robert Zimmermann, Taha Falatouri, Farzaneh Darbanian, and Tobechi Obinwanne. 2023. Applications of large language models (LLMs) in business analytics – exemplary use cases in data preparation tasks. In *HCI International 2023 – Late Breaking Papers* (Lecture Notes in Computer Science). Vol. 14059. Springer, 182–198. DOI: 10.1007/978-3-031-48057-7_12.

[15] Humam Kourani, Alessandro Berti, Daniel Schuster, and Wil M. P. van der Aalst. 2026. Evaluating large language models on business process modeling: framework, benchmark, and self-improvement analysis. *Software and Systems Modeling*, 25, 4, 1151–1186. DOI: 10.1007/s10270-025-01318-w.

[16] Jason Priem, Heather Piwowar, and Richard Orr. 2022. OpenAlex: a fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv: 2205.01833 [cs.DL].

[17] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. AutoGluon-Tabular: robust and accurate AutoML for structured data. arXiv: 2003.06505 [stat.ML].

[18] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 18932–18943.

[19] Anthropic. 2025. Claude Sonnet 4.5 System Card. Retrieved Sept. 5, 2026 from https://www.anthropic.com/claude-sonnet-4-5-system-card.

[20] OpenAI. 2025. Update to GPT-5 System Card: GPT-5.2. Retrieved Sept. 5, 2026 from https://openai.com/index/gpt-5-system-card-update-gpt-5-2/.

[21] Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39, 3/4, 324–345. DOI: 10.2307/2334029.

[22] Matjaž Gams. 2025. The oath of researchers and developers. *Informatica*, 49, 1, 1–6. DOI: 10.31449/inf.v49i1.8149.

Graphs of *n*-grams for Text Generation and Characteristic Phrase Extraction in the Domain of Smell

Nadezhda Komarova
Jožef Stefan Institute
Ljubljana, Slovenia
Technical University of Munich
Munich, Germany

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

We present a transparent, graph-based approach to text generation and characteristic phrase extraction in the domain of smell. Weighted directed graphs of *n*-grams are constructed from the smell-filtered portion of the Google *n*-grams dataset (20,000–60,000 *n*-grams), and three pathfinding strategies – a greedy algorithm, the Bellman–Ford algorithm, and topological sorting – generate continuations of olfactory seed phrases. As reference texts are unavailable, generation is evaluated with two graph-tailored metrics complemented by sentence-transformer coherence scores, and we study the effect of the seed *n*-gram degree, the dataset size, and whether the graph is cyclic. Building on the same structures, vertex-level clustering coefficients and closeness centrality extract phrases characteristic of olfactory subdomains such as floral scents, under three settings: default, verb-filtered, and generation-augmented. The two families of metrics are found to disagree – the greedy algorithm scores highest on the graph-tailored metrics while the global methods score highest on coherence – and the three extraction settings surface qualitatively different olfactory vocabulary, offering an interpretable complement to purely neural approaches.

Keywords

Smell, Olfactory Language, Graphs, *n*-grams, Clustering Coefficients, Centrality Measures, Text Generation, Keyphrase Extraction

1 Introduction

Olfactory language – the vocabulary used to describe scents, fragrances and odours – is notoriously difficult to process automatically. Unlike vision or sound, smell lacks a small, well-agreed vocabulary of primitives, and descriptions of scent are improvised and metaphorical: a perfume is *powdery* or *green*, an animalic note *sensual* in marketing copy and *dirty* in a review. Identifying characteristic olfactory phrases nevertheless has practical value: fragrance companies catalogue and search products by textual description, and recommender systems relate free text to scent families such as floral, musky or citrus.

Large language models generate fluent olfactory text but are opaque: it is hard to inspect *why* a phrase was produced, or to keep a description anchored to one scent family. This motivates interpretable, lightweight alternatives that explicitly model co-occurrence relationships between phrases and allow generation and extraction to be traced in a structured manner. This is knowledge discovery from a corpus: *n*-gram statistics are mined, represented as a weighted graph, and mined again with network measures.

This paper investigates *n*-gram graph structures as such an alternative. The first stage predicts a plausible continuation of an initial olfactory *n*-gram: graphs are built from co-occurrence statistics in the Google *n*-grams dataset [10], and prediction is formulated as graph traversal using a greedy algorithm, the Bellman–Ford algorithm and topological sorting. As no reference texts are available, we introduce graph-tailored metrics and complement them with sentence-transformer coherence scores. The second stage reuses the same construction to extract *characteristic phrases* of smell subdomains – floral scents, for example – from vertex-level clustering coefficients and closeness centrality, under three extraction settings.

The contributions are: (i) a transparent *n*-gram graph model for phrase prediction in the smell domain, evaluated across pathfinding strategies, seed degree, dataset size and graph structure; (ii) graph-tailored metrics for reference-free generation, complemented by embedding-based coherence; and (iii) a method for extracting characteristic phrases of olfactory subdomains using clustering coefficients and centrality.

2 Related Work

Text generation and n-gram models. Neural text generation has moved from recurrent networks to contextualised representations and transformers [7, 8]. Naive *n*-gram models remain attractive for their low cost and ease of domain adaptation, though their output quality is well below state-of-the-art transformers.

Recent work narrows this gap: Shareghi et al. [15] establish stronger *n*-gram baselines, Li et al. [6] enhance an *n*-gram basis with a neural model, and Ikegami et al. [4] combine *n*-grams with a recurrent network. Closest to our setting, Nagalavi and Hanumanthappa [12] predict newspaper article-block sequences from *n*-gram probabilities, and Dasgupta et al. [1] study higher-order *n*-grams with Laplace smoothing. We reuse such probabilities not for prediction but as *edge weights*, letting graph traversal extract further structure.

Graph-based approaches. Graphs are widely used as input structure for generation: Koncel-Kedziorski et al. [5] generate text from knowledge graphs with graph transformers, and Gao et al. [2] use graph-augmented encoders for RDF-to-text. Graph algorithms also serve linguistic tasks directly: Prabhumoye et al. [13] cast sentence ordering as topological sorting, and Tromp and Pechenizkiy [16] combine *n*-gram frequencies with a graph for language identification.

Clustering coefficients and phrase extraction. Clustering coefficients quantify connected triangles in a network, applied for instance to small-worldness in brain networks [9]. We use them alongside closeness centrality, both indicating vertex importance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SiKDD 2026, October 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.91

Our task extends keyphrase extraction – detecting phrases that summarise a document – from one document to an entire domain, here *smell*.

3 Methodology

An overview of the two stages is given in Figure 1. Both stages operate on the same weighted directed graph; the dashed arrow marks the generation-augmented setting, where Stage 1 output feeds Stage 2.

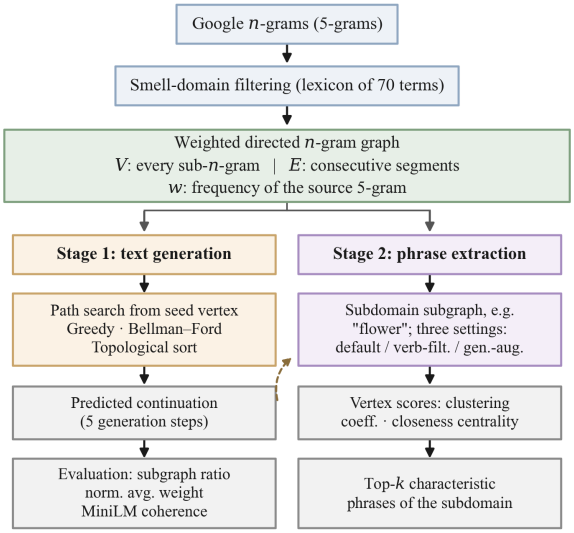


Figure 1: Overview of the methodology.

3.1 Stage 1: Text Generation

3.1.1 Constructing the graph. The graph is built from the smell-related 5-grams of the Google n-grams dataset. Each 5-gram is partitioned in every possible way, i.e. into every concatenation of 1-, 2-, 3-, 4- and/or 5-grams, each a contiguous substring of the initial 5-gram. Every such substring becomes a vertex of a directed graph, and an edge connects a substring to the one that follows it in the source 5-gram, weighted by the frequency of that 5-gram. Because substrings are shared across partitions, vertices are reused and the result is a genuine graph rather than a set of disjoint chains.

3.1.2 First approach: greedy algorithm. The generated text corresponds to a path through the graph. The algorithm starts at the vertex matching the *initial n-gram* to be completed, and at each step selects the locally optimal continuation, i.e. the outgoing edge of maximum weight. The number of steps equals the number of individual phrases to be predicted. Its weakness is that always selecting the locally optimal phrase may degrade quality for longer generated texts, which motivates the two global alternatives below.

3.1.3 Second approach: Bellman–Ford. This method finds the path between two vertices with maximal edge-weight sum. The Bellman–Ford algorithm, normally used for single-source shortest paths, is repurposed by multiplying edge weights by -1; for this to be correct, negative cycles must be eliminated. The graph is therefore replaced by its maximum spanning forest, the collection of maximum spanning trees. To generate text, the subgraph in the proximity of the source vertex is selected via depth-first search, containing all vertices whose shortest path from the source uses at most as many edges as the degree of the n-gram to be generated. Finding the spanning forest costs $O(m \log n)$ for n vertices and m edges. It is preferred over decomposition into strongly connected components ($O(n + m)$), which can leave semantically related concepts mutually unreachable. Spanning-forest construction discards information, but connectedness is itself of interest for the later centrality and clustering measures.

3.1.4 Third approach: topological sorting. This approach shares the intuition of the previous one. The maximum spanning forest is computed to eliminate negative cycles and the topological order of the vertices is obtained. Every vertex is assigned an initial distance of $-\infty$, except the source, which is assigned 0. Edges are then sorted ascending, first by the topological order of the origin vertex and then by that of the target vertex, and each edge is considered in turn for whether it improves the current distance to its target – the established technique for longest-path finding in a directed acyclic graph.

3.2 Stage 2: Characteristic Phrase Extraction

The second task is extracting phrases characteristic of a given subdomain of smell. To assess phrase significance and the semantic connections between expressions within a subdomain, clustering coefficients and centrality measures are computed for every vertex. Since the graph is weighted by n-gram frequencies, the clustering coefficient c_u of vertex u is defined as in [11],

$$c_u = \frac{1}{\deg(u)\big(\deg(u)-1\big)} \sum_{\substack{v,w \in N(u) \\ v \neq w}} \big(\widehat{w}_{uv}\widehat{w}_{uw}\widehat{w}_{vw}\big)^{\frac{1}{3}}, \qquad (1)$$

where $N(u)$ is the set of neighbours of u, $\deg(u)$ its degree, and $\widehat{w}_{uv} = w_{uv}/\max(w)$ the weight of the edge between u and v normalised by the maximum edge weight in the graph. The sum runs over ordered pairs of neighbours, so each triangle contributes twice.

3.2.1 Extraction procedure. Phrases whose vertices have higher clustering coefficients or centrality are assumed to be more characteristic of the subdomain. The procedure is: (1) select a list of words defining subdomains of smell, e.g. *musk*, *vanilla*, *flower*, *rose*, *salt*, *burnt*, *lemon*, *lavender*, *sulfur*, *perfume*, *vinegar*, chosen to span the perceptual families of perfumery and to vary in corpus frequency; (2) filter the data to n-grams containing that word and build a graph as in Section 3.1.1; and (3) compute clustering coefficients and closeness centrality per vertex, taking the top k as characteristic.

3.2.2 The three settings. Three variants of this procedure are compared. The **default** setting applies the procedure unchanged. The **verb-filtered** setting additionally requires each n-gram to contain a smell-related verb. The **generation-augmented** setting adds a preprocessing step in which each original n-gram is extended by the topological-sorting generator of Section 3.1.4, the extended n-grams replacing the originals before the graph is built.

Table 1: Text generation in the smell domain: subgraph ratio (SR) and normalised average weight (NW), aggregated over each factor in turn.

	SR	NW
Pathfinding approach		
Greedy	19.91	0.0303
Bellman–Ford	16.90	0.0312
Topological sorting	16.90	0.0312
Seed n-gram degree		
2	10.85	0.0186
3	15.82	0.0279
4	22.98	0.0356
5	26.75	0.0459
Graph type		
Acyclic	16.75	0.0301
Cyclic	23.38	0.0327
Dataset size (n-grams)		
20,000	16.45	0.0253
40,000	17.48	0.0286
60,000	20.80	0.0370

4 Experimental Settings

4.1 Stage 1: Text Generation

Prediction quality is compared across four models: greedy on the spanning forest (acyclic), greedy on the full graph (cyclic), Bellman–Ford, and topological sorting.

Metrics. The *subgraph ratio* is the mean edge weight along the predicted path divided by the mean edge weight over the whole subgraph defined by the number of prediction steps. The *normalised average weight* is the mean edge weight of the predicted path divided by the maximum edge weight in the graph. Coherence is measured as the cosine similarity between sentence embeddings of the generated text produced by the `all-MiniLM-L6-v2` sentence transformer [14].

Setups. All setups use 5 prediction steps. The effect of seed length is studied on sets of 100 randomly selected *n*-grams from the Google *n*-grams dataset consisting solely of 2-, 3-, 4- or 5-grams. Where a whole seed phrase was not present as a vertex, its longest contained suffix was used. The dataset was filtered to the smell domain using the expressions *stink, scent, scents, smell, smells, odour, odor, odours, odors, stench, reek, aroma, aromas, aromatic, whiff etc.* Words containing non-alphabetical characters were removed. The expressions are thesaurus synonyms of olfactory nouns, verbs and adjectives, with inflectional variants included so that frequency mass is not split across forms.

4.2 Stage 2: Characteristic Phrase Extraction

Subdomains are specified by restricting the dataset to *n*-grams containing the defining phrase, and characteristic phrases are those whose vertices have the highest clustering coefficients or closeness centrality; the top $k = 10$ are reported. Clustering coefficients and centrality are computed with NetworkX [3]. The verbs used for verb-filtering are *smell(s/ed/ing)*, *reek(s/ed/ing)*, *waft(s/ed/ing)*, *stink(s/ing)*, *stank*, *stunk* and *perfume(d/ing)*, on the intuition that the domain contains many *smelling like* constructions relating qualitatively different concepts.

5 Results and Discussion

5.1 Stage 1: Text Generation

Results are given in Tables 1 and 2. The graph-tailored metrics are identical for Bellman–Ford and topological sorting, as expected:

Table 2: Average coherence of generated text (cosine similarity between `all-MiniLM-L6-v2` embeddings), at seed degree 4 and 60,000 *n*-grams.

Graph type	Approach	Avg. coherence
Acyclic	Greedy	0.2175
Cyclic	Greedy	0.2230
Acyclic	Topological sorting	0.2365
Acyclic	Bellman–Ford	**0.2366**

both find the maximum-weight path from the same acyclic subgraph. Reporting both remains meaningful because they diverge under other cycle-elimination strategies. Cyclic graphs yield higher subgraph ratios than acyclic ones, though this may reflect that metric's sensitivity to edge count; the normalised average weight is comparable. The two global approaches consistently yield *lower* subgraph ratios than greedy search – counterintuitive, since greedy is the method prone to local optima, and possibly a property of this smell-filtered subset. Greedy is also the lightest computationally.

Seed length. Performance could not be measured for seeds absent from the graph; resolving them by semantic similarity to the nearest contained phrase would raise cost. High-weight paths are likelier from short seeds, since more vertices correspond to low-degree *n*-grams.

Dataset size. Both metrics increase with dataset size and are comparable at 40,000 and 60,000 *n*-grams. When the dataset was too small, few seeds had a large enough connected neighbourhood: generation steps could not be completed, or were confined to a cycle, suggesting many small strongly connected components.

Metric disagreement. The two families of metrics do not agree (Table 2). Greedy search scores highest on the graph-tailored metrics, yet Bellman–Ford and topological sorting achieve higher coherence. Globally optimal paths therefore yield more coherent output. The graph-tailored scores are functions of edge weight alone, i.e. of corpus transition frequency, and so indifferent to topical drift, near-duplicate wording and grammaticality – the properties coherence responds to.

Qualitative inspection indicated that the generated texts are not comparable in fluency to LLM output, motivating the shift to Stage 2, where the graph is used for extraction rather than generation. Even so, deterministic behaviour has value where a reproducible choice among near-synonymous descriptors is preferable to opaque generation.

Temporal composition. Because the frequencies are aggregate counts, the graphs describe olfactory language across the corpus as a whole rather than in any single period. For *flower* the phrases that surface are predominantly contemporary and commercial (Table 3), indicating where the frequency mass of the subdomain lies.

5.2 Stage 2: Characteristic Phrase Extraction

Table 3 presents results for *flower*. The separation into subdomains is not absolute; *subdomain* here denotes smells tied to a set of circumstances. Clustering coefficients and centrality select overlapping *n*-grams and appear comparably meaningful. The three settings surface different aspects of the subdomain.

Table 3: Top 10 phrases for the *flower* subdomain of smell, ranked by clustering coefficient (CC); three extraction settings.

Default		**Verb-filtered**		**Generation-augmented**	
n-gram	CC	*n*-gram	CC	*n*-gram	CC
Essential Oils Aromatic	0.0083	brazil flower fragrant perfume	0.0273	of this vivacious	0.0026
Fragrance Flower Fresh Flower	0.0075	smell the flowers along	0.0251	and decoration set	0.0019
Fresh Flower Ftd	0.0075	smell the flowers in	0.0233	I have ever	0.0014
Drawings Flower Essence Flower	0.0075	smell of wild flowers	0.0213	that grow only	0.0013
Essences Flower Fairy	0.0075	sweet smelling yellow flowers	0.0203	bloom from July	0.0013
Essence Flower Essences Flower	0.0075	smells of the flowers	0.0199	in late winter	0.0012
candles romantic	0.0065	smell the flowers on	0.0128	of the Gardenia	0.0012
fragrance is a	0.0060	sweet smell of flowers	0.0119	covering the entire	0.0012
and seasonal blooms	0.0058	the flowers smell and	0.0116	attract insects	0.0005
New Action	0.0058	smell the flowers or	0.0100	include freesia	0.0004

The **default** setting favours compound noun phrases tied to fragrance products. The **verb-filtered** setting targets the *"subject smells [like] object"* construction, recovering the *object* as a characteristic *n*-gram; it yields longer phrases but repeats near-identical variants ("smell the flowers along", "…in", "…on"). The **generation-augmented** setting, extending each *n*-gram by 5 generation steps before construction, produces more specific descriptors ("of the Gardenia", "include freesia"); one generation step did not change the top 10 appreciably. No setting is uniformly best: the default and verb-filtered settings might suit fragrance recommendation, the generation-augmented setting fine-grained vocabulary discovery – provided its output is validated. Numerical comparison with Section 2 is obstructed by the absence of a shared task: those methods are scored against held-out text or per-document gold keyphrases, neither available here.

6 Conclusion and Future Work

This paper explored graph algorithms for text generation and characteristic phrase extraction in the domain of smell using simple *n*-gram graph models. We examined the influence of dataset size, *n*-gram length, graph structure, and pathfinding algorithm on generation quality, and extracted phrases characteristic of olfactory subdomains (*flower*) under three settings. Human evaluation is the clearest next step: annotators rating whether each extracted phrase is characteristic of its subdomain, and a blind comparison of 20–30 continuations against LLM output. A frequency baseline should accompany it, with dispersion reported alongside the means, and a time-stamped corpus would allow the same construction per period. Reference-free metrics and a scalability analysis would clarify the interpretability–runtime trade-off [4]. Subdomains could be defined more flexibly through multi-term filtering or unsupervised graph clustering [17]. Applying the extracted phrases in fragrance recommendation or retrieval-augmented generation could further establish their practical value.

Acknowledgements

This work was supported by the Slovenian Research Agency under the project J7-50233, by the European Union through enrichMyData EU HORIZON-IA project under grant agreement No 101070284 and DataPACT EU HORIZON project under grant agreement No 101189771.

References

[1] Shibaranjani Dasgupta, Chandan Maity, Somdip Mukherjee, Rohan Singh, Diptendu Dutta, and Debasish Jana. 2024. HITgram: a platform for experimenting with *n*-gram language models. (2024). arXiv: 2412.10717 [cs.CL].

[2] Hanning Gao, Lingfei Wu, Po Hu, and Fangli Xu. 2020. RDF-to-text generation with graph-augmented structural neural encoders. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 3030–3036. doi:10.24963/ijcai.2020/419.

[3] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy)*, 11–15.

[4] Yukino Ikegami, Setsuo Tsuruta, Andrea Kutics, Ernesto Damiani, and Rainer Knauf. 2024. Fast ML-based next-word prediction for hybrid languages. *Internet of Things*, 25, 101064. doi:10.1016/j.iot.2024.101064.

[5] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2284–2293. doi:10.18653/v1/N19-1238.

[6] Huayang Li, Deng Cai, Jin Xu, and Taro Watanabe. 2022. N-gram is back: residual learning of neural text generation with *n*-gram language model. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 1523–1533. doi:10.18653/v1/2022.findings-emnlp.109.

[7] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: a survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4492–4499. doi:10.24963/ijcai.2021/612.

[8] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open*, 3, 111–132. doi:10.1016/j.aiopen.2022.10.001.

[9] Naoki Masuda, Michiko Sakaki, Takahiro Ezaki, and Takamitsu Watanabe. 2018. Clustering coefficients for correlation networks. *Frontiers in Neuroinformatics*, 12, 7. doi:10.3389/fninf.2018.00007.

[10] Jean-Baptiste Michel et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331, 6014, 176–182. doi:10.1126/science.1199644.

[11] Kent Miyajima and Takashi Sakuragawa. 2014. Continuous and robust clustering coefficients for weighted and directed networks. (2014). arXiv: 1412.0059 [cs.SI].

[12] Deepa Nagalavi and M. Hanumanthappa. 2016. N-gram word prediction language models to identify the sequence of article blocks in English e-newspapers. In *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, 307–311. doi:10.1109/CSITSS.2016.7779376.

[13] Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2020. Topological sort for sentence ordering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2783–2792. doi:10.18653/v1/2020.acl-main.248.

[14] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. doi:10.18653/v1/D19-1410.

[15] Ehsan Shareghi, Daniela Gerz, Ivan Vulić, and Anna Korhonen. 2019. Show some love to your *n*-grams: a bit of progress and stronger *n*-gram language modeling baselines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4113–4118. doi:10.18653/v1/N19-1417.

[16] Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *Proceedings of the 20th Machine Learning Conference of Belgium and The Netherlands (Benelearn)*, 27–34.

[17] Timothé Watteau, Aubin Bonnefoy, Simon Illouz-Laurent, Joaquim Jusseau, and Serge Iovleff. 2024. Advanced graph clustering methods: a comprehensive and in-depth analysis. (2024). arXiv: 2407.09055 [cs.LG].

QFS-Composer: Query-focused Summarization Pipeline for Less Resourced Languages

Vuk Đuranović*
vd0284@student.uni-lj.si
University of Ljubljana
Ljubljana, Slovenia

Marko Robnik-Šikonja*
marko.robnik@fri.uni-lj.si
University of Ljubljana
Ljubljana, Slovenia

Abstract

Large language models (LLMs) demonstrate strong performance in text summarization, yet their effectiveness drops significantly across languages with restricted training resources. This work addresses the challenge of query-focused summarization (QFS) in less-resourced languages, where labeled datasets and evaluation tools are limited. We present a novel QFS framework, QFS-Composer, that integrates query decomposition, question generation (QG), question answering (QA), and abstractive summarization to improve the factual alignment of a summary with user intent. We test our approach on the Slovenian language. To enable high-quality supervision and evaluation, we develop the Slovenian QA and QG models based on a Slovene LLM and adapt evaluation approaches for reference-free summary evaluation. Empirical evaluation shows that the QA-guided summarization pipeline yields improved consistency and relevance over baseline LLMs. Our work establishes an extensible methodology for advancing QFS in less-resourced languages.

Keywords

summarization, large language models, less-resourced languages, question answering based evaluation, Slovene

1 Introduction

In recent years, the world has witnessed rapid progress in generative artificial intelligence (AI) models capable of producing coherent, complete text. However, several works. e.g., [9] showed that LLMs perform better for languages that were highly represented in the training set, such as English, Chinese, and Spanish, compared to less-resourced languages. We aim to address this performance gap in query-focused summarization (QFS), also known as target-based summarization, a task in which the goal is to summarize a given text with respect to the query provided to the model. A common use case for QFS is in public relations and media press clipping, where companies are interested in how their firm is portrayed in a given text or how news may affect their business.

The crucial in our work is performance evaluation. For the summarization task, a common practice is to compare the LLM's summary with a reference summary created by qualified human experts. Although this approach has benefits when we want to train the model to respond in a specific style, the downside is that collecting reference summaries is time-consuming and expensive. Therefore, following prior work on less-resourced languages, we resort to automatic, reference-free evaluation metrics [11, 10, 6]. However, most of these evaluation metrics were developed for English, and we need to adapt them for our language of choice.

The contributions of this work are the following:

- Our own QFS system called ***QFS-Composer*** that is based on question answering[1].
- Working QG[2]and QA[3] models.
- Translation of MOCHA dataset [3] to the Slovenian language[4].
- Adaptation of QuestEval[5], QAGS (see footnote 1), and RQUGE[6] for evaluation of summaries in Slovene and comparison of summarization systems.

2 Related work

Query-focused summarization is highly overlapping with the question-answering (QA) task and is sometimes also referred to as factoid text summarization. A complementary task to QA is question generation (QG), where the model generates questions based on the observed document. A common approach is to pass the text to a named entity recognition (NER) model, which then detects and extracts named entities present in the text, and then those named entities are fed to a QG model that creates questions for which the extracted entity should be the correct answer.

Both QG and QA are used in our QFS approach as building blocks of summarization models and for summary evaluation. Fabbri et al. [4] provides a thorough analysis of the correlation between human quality judgments and automatic reference-based evaluation metrics for summarization, such as ROUGE [5] (and its many variations), BERTscore [15], and BLEU [7]. A QA-based evaluation technique called QAGS [11] showed a stronger correlation with human preference than standard evaluation scores. QuestEval [10], another evaluation approach based on question generation and answering that builds upon the QAGS framework, also showcased superior performance compared to ROUGE and other standard metrics when evaluated against reference summaries.

For automatic text summarization in Slovene, the *SloMetaSumm* [13] system recommends the most suitable summarization model for a given Slovene text. The development of these models was supported by datasets extracted from large Slovenian text corpora, such as KAS [14]. Regarding the work in QFS, to the best of our knowledge, the only existing work is pre-neural [2].

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).

[1]https://github.com/VukMNE/qfs-composer-paper
[2]https://huggingface.co/VukDju/GaMS-9B-Instruct-QG-Full-3ep
[3]https://huggingface.co/VukDju/GaMS-9B-Instruct-QA-3ep
[4]Dataset available at: https://github.com/VukMNE/MOCHA_slo
[5]https://github.com/VukMNE/QuestEval
[6]https://github.com/VukMNE/RQUGE

3 QFS-Composer

In this section, we present the main components of our QFS system, named QFS-Composer. We start with the Query Decomposer, followed by a QA model paired with a span scorer, and finish with an LLM that produces the final output. The main idea behind this architecture is to augment the original prompt with question-answer pairs and enforce factual consistency in model outputs.

3.1 Query Decomposer

Queries used in QFS can be broad, topic- or aspect-oriented, and sometimes more like a command or a request for information retrieval than a real question. Moreover, a large portion of queries require responses to multiple subtasks or sub-questions. A frequent request in QFS is a comparison or a summarization of a document from the perspective of an entity mentioned in the text. Hence, we believe that the best way to provide a summary that satisfies all requests from the user's query is to split the initial query into separate, smaller tasks. After successfully decomposing the query, we forward the questions to the QA model to find answers in the source text, and pass the user's query, along with extracted question-answer pairs, as an augmented prompt to the model, which will then generate a summary.

Ideally, we would train a separate supervised model for query decomposition using a dataset specifically dedicated to that purpose, as in the QDMR system [12]. Unfortunately, the QDMR query decomposition model was trained on the `BREAK` dataset, designed for a multi-document setting that does not align with the focus of our research. Since it is not expected that such datasets would exist for less-resourced languages (we could not identify any such dataset for Slovene), we propose two possible workarounds.

The first approach (called **`decomp_aug`** in the experiments in Section 4.3) is to utilize an LLM with a specially designed few-shot prompt to decompose the query into multiple sub-questions. A possible downside of this approach is that the LLM could introduce questions that are very different from the type of questions present in the SQuAD dataset (used to train our QA component), which could consequently lead to the QA model not being able to answer them. Nevertheless, we tried this approach, and the prompt we designed for an LLM to decompose the user query is available in Appendix A.

The second possible approach (called **`ner_aug`** in the experiments in Section 4.3), suitable for less-resourced languages, is to employ a named entity recognition (NER) model that scans the user's query and extracts entities for which our QG model subsequently generates questions. The downside of this approach is its limitation to named entities as the focus of summarization. As real-world applications of QFS mostly use named entities, this is an acceptable limitation. In Section 4, we test both approaches and report the results.

3.2 The QA component

An integral part of our summarization system is a question answering model. The purpose of this component is to emphasize the factual information in the source text relevant to the user's query. The QA component is used to answer the questions produced during query decomposition, and the answers, along with their accompanying questions, are merged with the user's query and source text into the augmented prompt sent to the LLM for summary generation.

As the base for our Slovene QA model, we utilized the top Slovene LLM, GaMS-9B-Instruct, and fine-tuned it for three epochs on the Slovenian translation [1] of SQuAD v2 dataset [8] using the token-level cross-entropy loss function. The initial learning rate α was set to 2×10^{-6}. The learning rate schedule employed a linear warmup for 10% of the total training steps, followed by linear decay. We used per-device batch size 1 with 16× gradient accumulation (effective batch size 16), AdamW (8-bit) as implemented in `adamw_bnb_8bit` (with weight decay set to 0.01 and $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$), and gradient-norm clipping set to 0.1. Training was performed in **BF16** mixed precision, and validation set evaluation was performed at the end of each epoch. The QA model was trained under the same computational setup, using a single NVIDIA A100 GPU (80 GB memory).

The integration of the QA model into our architecture is not straightforward due to the memory constraints of our hardware. Preliminary experiments with different length inputs resulted in frequent out-of-memory errors, so 350 characters was chosen as the maximum stable value across runs. This restriction means that the model without adjustments would not be able to answer a question when given a longer document as context, especially if the answer lies in the middle or at the end of the document.

To mitigate this issue, we split longer text into smaller chunks that the QA model can process. For each chunk, we compute BERTScore with respect to the question and sort the chunks in descending order by this score. We let the QA model answer only n top-ranked chunks for which the BERTScore is higher than the threshold value of 0.85, and we take the most frequent prediction in the selection as the final answer to the question. If there is a tie, the answer from the chunk with a higher BERTScore is selected as the output. The BERTScore threshold of 0.85 was selected heuristically during the development phase based on manual inspection of several unrelated question–chunk pairs: values below 0.85 often admitted chunks that were only loosely related to the question, whereas higher thresholds tended to filter out relevant chunks. A more systematic ablation of the BERTScore threshold and the choice of n is left for future work.

3.3 LLM for the final answer generation

The final component in our pipeline is the LLM model that performs the summarization on the previously obtained information. The idea is that this LLM will produce better summaries if we provide it with the augmented query containing subquestions extracted from the query and the answers to those questions provided by the QA model using the source document as context. The structure of the augmented prompt we send to the LLM is in Appendix A.

4 Evaluation

For less-resourced languages, evaluation methods that rely on reference summaries are usually off the table. Therefore, we resort to methods based on automatic question generation and answering, and first describe the adaptations of QuestEval [10] and QAGS [11] to Slovene. Next, we describe our experimental settings and results. We end with the qualitative analysis of the results.

4.1 QAGS

The first step in the QA evaluation framework QAGS [11] is to extract K named entities from a summary. For each extracted named entity, the QG model forms questions such that the named entity

is the correct answer to the generated question, conditioned on the summary. Next, the QA model answers the questions generated in the previous step, but it answers them conditioned on the source text. After that, another set of answers is generated by the same QA model but using the candidate summary as the only context. Finally, a similarity metric is computed between pairs of answers for every question, and their average is the returned score of the metric. In the original paper, authors have used both token-level F1 and Exact match score as similarity measure between two sets of answers.

4.2 QuestEval

The QuestEval [10] evaluation framework, in contrast to QAGS, generates two sets of questions instead of one. The first set of questions is created in identical manner as is the case in the QAGS methodology, by extracting named entities from the candidate summary and forming questions around them. The process continues with the QA model generating two answers for each question, one that is conditioned on the summary and the other based on the entire source document. The second set of questions is generated from the entire original text, and the QA model then generates answers for each question by alternating between the candidate summary and the source document as the only context. The purpose of this additional set of questions is to evaluate the summary based on whether it captivates important information from the whole document.

The novelty in the QuestEval framework is the introduction of the Weighter component that performs the task of assigning weights to each question generated from the original document based on its relevance to the text. The Weighter component enables QuestEval to factor in the amount of crucial information from the source that is present in the summary. Additionally, the formula for computation of the QuestEval score features the Confidence of answerability, which represents the probability of the QA model not predicting a special non-answerable token as an answer to a specific question. In this way, if the QA model deems important questions from the source that have answers within the document unanswerable when using the candidate summary as the only context, the summary will be assigned a lower score. Finally, the output of the QuestEval is the F_1 score, where the precision component measures the consistency of the summary with the source text by comparing whether answers to the first set of questions differ when the QA model is conditioned on the summary versus the original document. The recall component of QuestEval assesses whether the summary contains the most important information from its source text by comparing the answers to the second set of questions.

4.3 The experimental setting

To evaluate the effectiveness of our approach, we selected four LLMs to perform query-focused summarization. For every LLM selected as a summarizer model in our system, we utilize three different settings (see Section 3.1):

- **`decomp_aug`** - using another LLM to decompose the query into sub-questions that will be answered by the QA component and in that way augment the prompt;
- **`ner_aug`** - NER model searches for named entities within the query, and forwards them to our QG model that creates questions using them;
- **`no_aug`** in which the user's query is passed straight to the LLM as it is.

We use the following LLMs: GPT-4.1-mini, Gemma 2 9B-It, Llama 3.1 8B Instruct, and GaMS 9B Instruct.

For the purpose of this evaluation, we collected a set of 21 texts in Slovene from popular Slovenian web news sources (delo.si, dnevnik.si, 24ur.com, siol.net). The summaries generated by models for each setting are first evaluated using the QAGS and QuestEval frameworks and later manually analyzed. While the number of texts is low, it is sufficient to demonstrate quantitative differences and enable in-depth qualitative analysis.

4.4 Quantitative evaluation

The results of the automatic evaluation are discussed below: for QAGS, they are presented in Table 1, while the QuestEval scores are available in Table 3, within Appendix A. The qualitative analysis follows in Section 4.5. The core of QA-based methods is to measure the similarity between answers generated using the entire original document as context and answers produced using the candidate summary as only context. In this work, the similarity between answers is measured using four different metrics: Exact match score (EM), F_1-score, Edit Distance, and BERTScore F_1.

Table 1: Evaluation with QAGS framework across models and settings, presenting the mean ± standard deviation. The best score across all models and metrics is in bold, and the best result per model is highlighted with underline dashes.

Model	QAGS			
Setting	F_1	EM	Edit Distance	BERTScore$_{F_1}$
gpt-4.1-mini				
no_aug	0.164 ± 0.31	0.105 ± 0.31	45.505 ± 24.37	0.685 ± 0.36
decomp_aug	0.253 ± 0.39	0.2 ± 0.4	40.436 ± 27.23	0.732 ± 0.33
ner_aug	0.279 ± 0.42	0.245 ± 0.43	42.436 ± 29.81	0.699 ± 0.37
gemma-2-9b-it				
no_aug	0.19 ± 0.35	0.133 ± 0.34	46.362 ± 25.69	0.614 ± 0.4
decomp_aug	0.221 ± 0.36	0.152 ± 0.36	41.952 ± 24.45	0.756 ± 0.29
ner_aug	0.219 ± 0.35	0.143 ± 0.35	44.695 ± 24.28	0.751 ± 0.3
Llama-3.1-8B-Instruct				
no_aug	0.285 ± 0.37	0.162 ± 0.37	40.048 ± 24.4	0.773 ± 0.3
decomp_aug	0.251 ± 0.35	0.133 ± 0.34	40.695 ± 23.83	0.738 ± 0.33
ner_aug	**0.376 ± 0.43**	**0.286 ± 0.45**	36.724 ± 27.81	**0.815 ± 0.26**
GaMS-9B-Instruct				
no_aug	0.221 ± 0.33	0.105 ± 0.31	45.933 ± 23.47	0.672 ± 0.37
decomp_aug	0.251 ± 0.36	0.162 ± 0.37	39.886 ± 25.22	0.739 ± 0.32
ner_aug	0.322 ± 0.4	0.209 ± 0.41	**36.305 ± 27.65**	0.784 ± 0.29

The results show that Llama-3.1-8B-Instruct dominates in F_1, EM, and BERTScore F_1, and GaMS-9B-Instruct produces the lowest Edit Distance score.

4.5 Qualitative analysis

Analysis of numerical results from QAGS and QuestEval metrics alone is insufficient to describe the overall behavior of the proposed QFS-Composer approach. Therefore, in this section, we take a closer look at summaries produced with practically the most important `ner_aug` setting of best-performing models **GaMS-9B-Instruct** and **Llama-3.1-8B-Instruct**. We compare them to responses from the same models when there is no prompt

augmentation. The texts, produced summaries, and prompts are available in our GitHub repository[7].

Upon examining summaries, we confirm that, in general, summaries generated with an augmented prompt are shorter, but more concise and information-dense than those generated with the plain approach. Notably, for queries in which the user wants to discover all instances of a certain phenomenon, or all factors that influence a certain thing, the basic no_aug approach yields a response in the form of an ordered list, while the augmented prompt approach yields plain text as output.

We observe for **GaMS-9B-Instruct** that if a named entity is omitted from the query, it is less likely that the same entity will be mentioned in the summary. This claim is best supported by summaries for text with ID #2, which correctly respond to the query but lack informative named entities indicating who performed the experiment and where it was conducted.

A review of augmented prompts utilized by models shows the limitations of the QA model. In certain cases, the QA model could not find an answer to a question, even though the answer clearly exists in the source text. The reason is that the QA model cannot handle long texts in one pass; instead, we split the document into chunks and rank them by BERTScore, which is computed by comparing each chunk to the question. Hence, it is possible that a chunk not containing the answer is ranked the highest, which can lead to no output or wrong answers from the QA model. We observed the same problem in the evaluation phase.

While manually inspecting model outputs, we found instances of LLM hallucinations. We observe that for the text in the interview form (text with ID: 10), model **GaMS-9B-Instruct** generated an off-topic response, while **Llama-3.1-8B-Instruct** hallucinated and made claims that are completely opposite to the content in the source text. All materials used in this experiment, including the source texts, generated summaries, and queries, can be found, as mentioned earlier, in our GitHub repository (link availalble in the footnote).

5 Discussion and conclusion

We introduced a novel framework for query-focused summarization, called **QFS-Composer**, and by leveraging query decomposition and intra-document information retrieval using our question generation and answering models, we found some indication of improved factual consistency and relevance of generated summaries compared to strong LLM baselines, although this finding is based on a small sample.

Beyond experimental performance improvements, a central contribution of this work lies in the development of new assets in the domain of the Slovenian NLP: QA and QG models optimized for factual supervision, translation of the MOCHA dataset, and adapted metrics such as QAGS, QuestEval, and RQUGE that are designed for reference-free summary evaluation. By combining these components within our framework, we have paved the way for further progress in the summarization process for less-resourced languages. Additionally, by making these resources modular and reusable, we aim to encourage future research and downstream benchmarking.

While our results show measurable gains, several challenges remain. First, our approach relies on query decomposition and QA model accuracy, so any errors at these stages propagate to the final task of summarization. Therefore, the future work should focus on improving the precision of each of the sub-components of QFS-Composer. Furthermore, because of hardware constraints, our QA and QG models are able to process limited amounts of text, which consequently forced us to split the original document into smaller text segments. A viable path for future research should explore whether the performance of our framework would improve if we could increase the amount of context that models can handle at once. Furthermore, future work should focus on collecting human-rated summaries generated with QFS-Composer and comparing them with summaries produced directly from user queries, to better quantify perceived usefulness and alignment with user expectations.

[7] https://github.com/VukMNE/qfs-composer-paper

Acknowledgments

This research was funded by the European Commission through grants 101186647 (AI4DH), 101254461 (SLAIF) and 101293013 (ELEXAI), and the Slovenian Research and Innovation Agency (ARIS) projects GC-0002, L2-50070, and core research programme P6-0411.

References

[1] Mladen Borovič et al. 2022. Slovene translation of the SQuAD2.0 dataset. Slovenian language resource repository CLARIN.SI. (2022). http://hdl.handle.net/11356/1756.

[2] Ercan Canhasi and Igor Kononenko. 2014. Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert systems with applications*, 41, 2, 535–543.

[3] Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. MOCHA: a dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (Nov. 2020), 6521–6532.

[4] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9, 391–409.

[5] Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, (July 2004), 74–81. https://aclanthology.org/W04-1013/.

[6] Alireza Mohammadshahi, Thomas Scialom, Majid Yazdani, Pouya Yanki, Angela Fan, James Henderson, and Marzieh Saeidi. 2023. RQUGE: reference-free metric for evaluating question generation by answering the question. In *Findings of the Association for Computational Linguistics: ACL 2023*, 6845–6867.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 784–789.

[9] Nathaniel Robinson, Perez Ogayo, David R Mortensen, and Graham Neubig. 2023. ChatGPT MT: competitive for high-(but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation*, 392–418.

[10] Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. QuestEval: summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6594–6604.

[11] Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[12] Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: a question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8, 183–198. Mark Johnson, Brian Roark, and Ani Nenkova, editors.

[13] Aleš Žagar and Marko Robnik-Šikonja. 2023. One model to rule them all: Ranking Slovene summarizers. In *International Conference on Text, Speech, and Dialogue*, 15–24.

[14] Aleš Žagar et al. 2022. Corpus of academic slovene KAS 2.0. Slovenian language resource repository CLARIN.SI. (2022). http://hdl.handle.net/11356/1448.

[15] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Experiment results

This section contains results of the summary evalaution experiment that we could not fit to the main article body. Table 2 shows average character and word lengths of summaries generated by models for each setting. Table 3 shows evaluation of LLMs for each experiment setting according to QuestEval evaluation framework.

Table 2: Character-wise and word-wise length of summaries generated by different models and settings.

Model Setting	Character count	Word count
gpt-4.1-mini		
no_aug	2080.48 ± 954.39	284.19 ± 126.23
decomp_aug	758.22 ± 214.46	103.95 ± 29.38
ner_aug	850.09 ± 308.53	117.5 ± 41.81
gemma-2-9b-it		
no_aug	1256.61 ± 352.64	177.91 ± 47.04
decomp_aug	555.67 ± 215.88	78.14 ± 28.98
ner_aug	578.33 ± 188.37	82.76 ± 26.23
Llama-3.1-8B-Instruct		
no_aug	1327.62 ± 137.52	196.28 ± 20.80
decomp_aug	1075.19 ± 318.52	157.24 ± 47.76
ner_aug	1157.48 ± 258.18	171 ± 40.04
GaMS-9B-Instruct		
no_aug	1016.91 ± 450.55	145.57 ± 63.19
decomp_aug	628.47 ± 257.39	91.81 ± 40.41
ner_aug	610.14 ± 263.57	89.81 ± 36.79

Table 3: Evaluation with QuestEval framework across models and settings presenting the mean ± standard deviation. The best score across all models and metrics is in bold, and the best result per model is highlighted with underline dashes.

Model / Setting	QuestEval (mean ± std)
gpt-4.1-mini	
no_aug	0.351 ± 0.109
decomp_aug	0.329 ± 0.113
ner_aug	0.332 ± 0.112
gemma-2-9b-it	
no_aug	0.363 ± 0.087
decomp_aug	0.363 ± 0.099
ner_aug	0.374 ± 0.126
Llama-3.1-8B-Instruct	
no_aug	0.381 ± 0.119
decomp_aug	0.348 ± 0.141
ner_aug	0.389 ± 0.131
GaMS-9B-Instruct	
no_aug	0.386 ± 0.134
decomp_aug	0.364 ± 0.150
ner_aug	**0.427 ± 0.141**

B Prompt templates

B.1 Query-decomposition prompt template

```
Razbij dano uporabniko poizvedbo v ve manjih,
↪  atomarnih podvpraanj, ...
- Vrni rezultat IZKLJUNO kot veljaven JSON objekt z
↪  ENIM poljem:
  "podvprasanja": ["...", "...", "..."]

### Primer 1
Uporabnika poizvedba: "Generiraj povzetek o porastu
↪  primerov garij v Sloveniji in svetu."
Podvprašanja (JSON objekt):
"podvprasanja": [
    "Koliko primerov garij je bilo v Sloveniji
    lani?",
    "Koliko primerov garij so v Sloveniji
    obravnavali letos?",
    "Koliko ljudi z garjami je Svetovna zdravstvena
    organizacija ocenila leta 2017?",
    "Koliko ljudi z garjami je Svetovna zdravstvena
    organizacija ocenila lani?",
    "Kateri so glavni razlogi za porast garij,
    navedeni v besedilu?"
    ]

### Primer 2
Uporabnika poizvedba: "Generiraj povzetek o tem,
↪  kako se garje irijo in zakaj zdravljenje pogosto
↪  spodleti."
Podvprašanja (JSON objekt):
"podvprasanja": [
    "Kako se garje prenašajo med člani
    gospodinjstva?",
    "Katere navade ali napake pri terapiji
    prispevajo k neuspehu zdravljenja?",
    "Kdo je po besedilu najbolj izpostavljen
    tveganju za okužbo z garjami?",
    "Kateri simptom je tipičen za garje ponoči?"
    ]

### Primer 3
Uporabnika poizvedba: "Generiraj povzetek navodil za
↪  zdravljenje garij."
Podvprašanja (JSON objekt):
"podvprasanja": [
    "Kako je treba nanašati kremo na telo pri
    odraslih?",
    "Kako pogosto in v kakšnih intervalih je treba
    ponoviti nanos zdravila?",
    "Kaj je treba storiti z oblačili in posteljnino
    pred in po terapiji?",
    "Na katerih mestih na telesu odraslih so
    spremembe najpogosteje opazne?"
    ]

### Nova uporabniška poizvedba:
"query"

Podvpraanja (JSON objekt z enim poljem
↪  "podvprasanja"):
```

B.2 QFS augmented prompt

```
[USER QUERY]
--------------------------
Besedilo: [SOURCE DOCUMENT]
--------------------------
V nadaljevanju so odgovori na vpraanja, povezana z
↪  poizvedbo:

Vprašanje: [QUESTION]
Odgovor: [ANSWER]
Vprašanje: [QUESTION]
Odgovor: [ANSWER]
...

Na podlagi zgornjih odgovorov, na kratko in jedrnato
↪  povzemite bistvo besedila v povezavi z zaetno
↪  poizvedbo.
```

Beyond Named Citations: A Corpus of Resolved Cross-References in EU Legislation

Jakob Vokač
jakob.vokac@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

EUR-Lex and its repository CELLAR make EU legislation freely available and machine-retrievable, but the cross-reference metadata they expose for legislation is essentially document-level and incomplete. Many legal automation tasks instead need each reference resolved to the provision it points to. We introduce an annotation scheme that pairs each reference, as a span in an act's text, with its target provision, and a ground-truth corpus of about 4,500 annotations over four EU acts, built by manually correcting the output of a mostly deterministic extraction pipeline. Measured against this corpus, two maintained EU linkers, the Commission's Ref2Link and LegalViz.EU, resolve the references they find well but miss most mentions, in particular indirect references that do not name their target. We also fine tuned and test a RoBERTa-large model, on our corpus using a cross-validation approach. It reached a high F1 score for detecting references on the validation sets. The dataset and code are available at https://github.com/JakobVokac/EURLex_CrossReferenceDataset_SiKDD2026.

Keywords

legal informatics, EU legislation, named entity recognition, entity linking, annotated corpus

1 Introduction

This work is one part of a larger effort to convert EU legislation into formal representations, in three stages: a structural knowledge graph of the legal text, whose nodes are the act's structural units and whose edges are the references between them; an ontology derived from the text; and a first-order logic world model built from the graph as source and the ontology as grounding. The motivation is to explore how different representations of legal text help an LLM agent acting as a legal assistant produce trustworthy and traceable reasoning paths, following previous work on building world models with LLMs in the educational domain [20]. The first stage is the problem of resolving cross-references: a reference reading *point (a) of Article 6(1)* is not an edge in any graph, and cannot be formalized or retrieved on its own, until that phrase is replaced by an unambiguous address of the unit it designates.

EU legislation is a favorable setting. CELLAR, the repository behind the EUR-Lex database of European Union law, serves every act as Formex, the XML dialect EUR-Lex uses to structure legal texts, whose markup fixes the units a reference can address — article, paragraph, subparagraph, point, annex — so the target space is closed and a resolution can be scored exactly rather than judged. Formex does not, however, resolve the references themselves. Prior work resolves legislative references in other jurisdictions, and two maintained linkers do so for EU law; none produces what this effort needs (see Related Work). Near-total reliability is nonetheless reachable, because both halves of the problem are constrained: Formex gives a well-defined structure to address, and EU drafting follows codified citation conventions [11, 24]. What remains is a residual of forms those conventions do not cover.

While one might be tempted to simply use an LLM agent with the proper tooling for finding references and retrieving the appropriate pieces of text, however, that approach is arguably unsuitable for the scale of this task. Querying CELLAR shows EUR-Lex holds roughly 1 million legislation documents documents and further documents covering individual legal disputes that might be applicable to individual interpretations of a legal problems, and the four documents covered in this work span already roughly ~4.5 thousand references. As such one can reasonably assume the total number of references spanning EUR-Lex would be in the hundreds of millions. Finding, disambiguating and retrieving the span of text for each such reference through an LLM approach would likely be prohibitively expensive.

As such, a more scalable approach must be devised. The work described in this paper makes the following contributions towards this:

(1) **An annotation scheme** pairing each mention, as a character span in the act's parsed source text, with a canonical target — a CELEX key, the identifier for most documents on EUR-Lex, and a subdivision path in the style of the European Legislation Identifier (ELI) — and the conventions the hard constructions force: conjunction under a shared qualifier, expanded ranges, several targets on one span, unnamed subparagraph steps.
(2) **A ground-truth corpus** of ~4,5k annotations carrying ~5,5k targets over four acts (GDPR, DSA, AI Act, SSM Framework Regulation).
(3) **A construction method built for expansion**: a staged extraction pipeline, mostly deterministic with LLM assistance for edge cases covering 0.35% of references, whose output a human reviews and corrects until a ground truth is reached.

2 Related Work

No freely available dataset publishes references and targets as defined in Section 3: mentions anchored as spans in the act's own text, every occurrence annotated, each resolved to a sub-document address. We are aware of no published evaluation of any system that extracts them. Two rule-based linkers come closest, and both compute what this dataset annotates without releasing it. Measured against the ground truth (Section 3) introduced here, they resolve direct mentions well and largely miss

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).

Table 1: Comparison of existing datasets and methods for annotations of references in legal text to other documents or parts thereof (cross-references). Corpus - what set of documents the work covers. Reference classes covered - types of references covered (internal, external, document vs. sub-document, etc..). Annotations released - resulting dataset size.

Resource	Corpus	Reference classes covered	Annotations released	Ref.
Annotated resources				
CDM / ELI metadata	EU, all of EUR-Lex	act-to-act citation edges	targets only, no mentions	[16, 23]
Legal NER corpora	DE, BR, IN, US, UK	reference strings, unresolved	8.8k–93k spans, no targets	[3, 13, 14, 17, 21, 27]
LegalNERo	RO national	named citations	3,387 spans, document-level ELI	[22]
Legal Linking	US constitutional	named and short-form cites	1,241 paragraphs, provision targets	[29]
Darji et al.	DE court decisions	statutory citations in judicial prose	2,944 records, subdivision targets	[7]
LegalPincite	EU case law (CJEU)	case- and paragraph-level citations	paragraph-level links	[25]
Extraction methods				
National resolvers	NL, LU, JP, GR, DE	named citations, internal structural	none	[1, 2, 4, 6, 8, 26, 30]
Ref2Link	EU, 24 languages	direct mentions; orphan citations listed as future work	none	[9, 10]
LegalViz.EU	EU, EUR-Lex	direct and bare internal; no anaphora, pro-forms or insertion context	none	[15]
This work				
Proposed corpus	EU legislation, 4 acts	direct **and** indirect: anaphora, pro-forms, defined short names, positional, insertion context	**4,402 spans, 5,452 sub-document targets**	

indirect ones (mentions that do not name their target in their own wording, but derive it from the surrounding context), which account for roughly 75% of our annotations. Table 1 shows the succinct picture.

EU citation networks draw their edges from cataloged metadata rather than from text [16, 23]. The Common Data Model (CDM) behind CELLAR does carry sub-document detail, but it is a case-law product: of 1.34M such annotations only 898 have a legislative citing work, and none of the 162 citation edges of the four acts studied here carries one. It is incomplete at document level too: its 64 outgoing citations for the AI Act reach 47 of the 187 distinct external acts the text cites.

Sub-document resolution from legal text with target resolution does exist, but only for national law [1, 2, 4, 6, 8, 26, 30]. All of it resolves by deterministic construction from the citation's surface form (instrument type, year, number), which by design never meets a reference whose target the wording omits.

Released datasets resolve references, or cover EU legislative text, but never both. The closest comparable work decomposes German references into 21 resolved fields, but is drawn from court decisions [7]; the larger legal named-entity-recognition (NER) corpora make a law or statute type their largest class [3, 13, 14, 17, 21, 27]. EU sub-document work exists on case law [25, 28], both excluding references to legislation, or hand-built [12]. The nearest adjacent work covers the same three regulations one layer up, extracting deontic obligations into graphs that presuppose resolved targets [5].

Two maintained linkers compute what this dataset annotates, and publish neither the spans nor an evaluation. Ref2Link (Commission Legal Service, EUPL-1.2) emits a CELEX, an ELI subdivision URL and a character offset [9, 10]. LegalViz.EU (Maastricht Law & Tech Lab, GPL-3.0) parses the same Formex source [15]. Tested against our dataset, both fall well short on detection, as Section 4 shows.

3 Annotation Scheme and Corpus

For our dataset annotation scheme we require clear definitions of what constitutes a reference and what its target is; together they fix exactly what the corpus annotates and what it leaves out.

Reference. A reference is a mention that explicitly designates one or more units of legislative text (a whole act, or a part, title, chapter, section, article, paragraph, subparagraph, point or annex within one) whether or not the wording identifies which unit; indirect mentions such as `that Directive` or `Article 4(2) thereof` are references too. We annotate the mention as a character span in the act's own parsed source text, and we annotate every occurrence, not only each distinct unit referred to. A mention extends across conjoined units only where a shared qualifier governs all of them: `Article 5 and Article 6 of that Directive` is one mention, since `of that Directive` distributes over both, whereas `point (a) of Article 6(1) or point (a) of Article 9(2)` is two, since each head carries its own qualifier. A mention begins at the unit designation, not at the words describing its content: in `the requirements laid down in Article 35(2) of Directive 2013/36/EU`, the mention is `Article 35(2) of Directive 2013/36/EU`.

Not references. A mention of content that designates no unit (e.g. `..the rules on data portability..`) is not a reference, nor is a mention of a body rather than a text, such as `the European Council`. We also leave out structural text such as labels, headings and tables of contents.

Target. The target is the canonical address of the unit designated: a CELEX key with an ELI-style subdivision path, as in

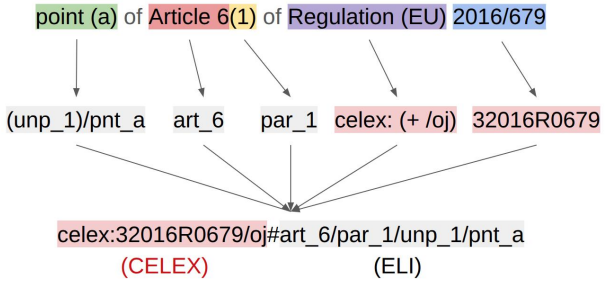


Figure 1: Illustrative example of a reference converted to a target.

Figure 1. The key identifies the act; `/oj` selects the version as published in the Official Journal rather than a later consolidated one; and the path names one structural unit per step, including steps the wording leaves implicit. A mention naming several units carries several targets on one span, and ranges are expanded. Instruments outside EUR-Lex are keyed to another authority, as `ext:std:iso-iec-17065`.

Corpus. The corpus covers four regulations — the GDPR, the Digital Services Act, the AI Act and the SSM Framework Regulation — carrying 4,509 annotated mentions and 5,452 targets (942, 1,167, 1,888 and 512 mentions per act, in that order); 451 mentions carry more than one target, and 5 targets are left explicitly unresolved. 4,402 mentions are anchored by character offsets; the remaining 107 were recorded by the reviewer without a span and are excluded from the span-level label set.

Construction. Annotation follows a recommend–revise scheme. A staged extraction pipeline proposes every candidate mention together with its target, and one human reviewer accepts, re-targets, rejects or adds, working in an interface that renders the act as published. The ground-truth is final pipeline independent file that correctly annotates all references in the review document.

Pipeline. The pipeline uses Formex as its input. Stage A parses Formex into a typed tree of citable units with ELI-style ids, and Stage B is a deterministic grammar over that immutable text; Stage C binds anaphora and pro-forms, and Stage D resolves footnote citations to a CELEX key. Every target any stage emits is validated against the registry built from the act's own parse, so a proposal addressing no unit that exists is never shown. Model calls are confined to two narrow jobs: binding bare anaphors, and one tier of footnote resolution.

Limits. There was only one annotator, who is not a legal expert, for the entire dataset. And because the reviewer works from proposals, a reference no rule proposed enters the corpus only if the reviewer spots it unaided; the 107 span-less annotations, 2.4% of the total, fall under that category.

4 Evaluation

Protocol. Every system is scored by one procedure against one gold set: the 4,402 mentions carrying character offsets, across all four acts. A gold mention counts as found when a system predicts at least one non-whitespace character inside it (boundaries vary by system and whether the mention span is fully identified is relevant, and expressed, in the target resolution accuracy). Identical prediction spans count once, and target accuracy is computed only over the mentions a system found, so detection and resolution do not confound. Neither linker has a notation for the unnumbered subparagraph step, so the subpath comparison tolerates it. One class of references, descriptive mentions, is held out of the scored export, and no figure here covers it.

Table 2: Mention (reference) detection (P - precision, R - recall, F1 - score) over the ground-truth references & target accuracy over the references each system found (where applicable).

System	P	R	F1	Target
Rule-based linkers				
Ref2Link	0.890	0.206	0.335	0.791
LegalViz.EU	0.765	0.406	0.531	**0.881**
Lexemo RoBERTa-large				
zero-shot	0.859	0.758	0.805	*N/A*
fine-tuned	0.946	0.993	**0.969**	*N/A*

Detector baseline. To test whether the corpus supports a learned fallback for acts no grammar has been written against, we fine-tune a token classifier on it. Lexemo's `roberta_large_legal_act_extraction` [18] is the only public checkpoint we found that extracts legal acts and articles from English EU legislative text; it is RoBERTa-large [19] and already task-adapted, so the zero-shot row is a strong baseline rather than a naive one. The original model had a 23-label head, which we collapsed to a single *detected* class. The fine-tuned model was trained and evaluation with cross-validation (3 training, 1 validation document split).

Reference detection. Fine-tuning on the corpus significantly improves the RoBERTa model performance. Mention F1 rises from 0.805 to 0.969. Additionally, fine-tuning the model raises the number of detected references that are correctly spanned from 13.5% to 94.5%. Ref2Link [9, 10] is an external-citation linker by design, and on external citations its recall (0.732) is ahead of LegalViz's (0.677). When accounting for all (internal and external) references in the proposed corpus, its recall drops significantly as shown in Table 2. This is because 72.5% of references in the corpus are internal. LegalViz [15] performs better on internal citations, yet still detects only 40.6% references overall.

Target resolution. On target resolution accuracy, both linkers are strong, and on external citations nearly identical: LegalViz 0.786, Ref2Link 0.789. LegalViz's higher performance in Table 2 is primarily due to its ability to detect internal references. The largest gap this corpus exposes for these two methods is on detection, not resolution.

Pipeline The dataset construction pipeline cannot be directly compared to any of the previous methods as it was built on the dataset documents themselves. The pipeline, when run on the corrected ground-truth gives: without the LLM feature — an F1 score of 0.985, and 0.999 target resolution; with the LLM feature — an F1 score of 0.986, and 0.998 target resolution.

5 Conclusion

We present an annotation scheme for references in EU legal documents, pairing each mention with a canonical target at sub-document granularity. Its purpose is to create a structural knowledge graph that keeps the text entire, connected along the references running within and between acts. We know of no existing resource that releases such annotations, or an evaluation over them.

We built a ground truth to that scheme by recommend–revise: a staged, mostly deterministic pipeline proposes every mention and its target, and a human reviewer accepts, corrects or adds. It spans four acts — GDPR, DSA, AI Act, SSM Framework Regulation — and holds 4,509 annotated mentions carrying 5,452 targets, available on [https://github.com/JakobVokac/EURLex_CrossReferenceDataset_SiKDD2026].

Measured against the corpus, the two maintained linkers find 20.6% and 40.6% of the annotated mentions, because indirect and act-internal references lie outside what they attempt; on the target resolution they both perform well. Fine-tuned on the corpus, a RoBERTa-large classifier finds 99.3% of the mentions in acts held out of its training. As such, given a large enough dataset, it can provide a better alternative for linking EUR-Lex legislation. That being said, a separate model would have to be trained specifically for target resolution, whose performance would have to at least match the existing linkers to make the entire approach a suitable alternative. Another caveat worth mentioning is that the documents selected are all relatively recent additions and mostly in the same area (apart from the SSM Framework Regulation). As such, the referencing styles are likely to coincide heavily, making overfitting a possible (at least partial) explanation for such good performance from the trained model.

Four acts are too few to predict how a method behaves across EUR-Lex, and the annotations carry one reviewer's judgment. Future work is to expand the corpus to a size on which a general method can be trained to an accuracy reliable enough for real users.

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency through the research program Knowledge Technologies (P2-0103) and the European Commission under the Horizon Europe project DataPACT, Grant Agreement No. 101189771. We would like to express our gratitude to all project partners for their contributions and collaboration.

References

[1] Morayo Adedjouma, Mehrdad Sabetzadeh, and Lionel C. Briand. 2014. Automated detection and resolution of legal cross references: approach and a study of Luxembourg's legislation. In *Proc. 22nd IEEE Int. Requirements Engineering Conf. (RE)*, 63–72.

[2] Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. 2018. Named entity recognition, linking and generation for Greek legislation. In *Legal Knowledge and Information Systems (JURIX 2018)*. IOS Press, Amsterdam, The Netherlands.

[3] Ting Wai Terence Au, Vasileios Lampos, and Ingemar Cox. 2022. E-NER — an annotated named entity recognition corpus of legal text. In *Proc. Natural Legal Language Processing Workshop (NLLP) at EMNLP*. Abu Dhabi, UAE, 246–255.

[4] Ilias Chalkidis, Charalampos Nikolaou, Panagiotis Soursos, and Manolis Koubarakis. 2017. Modeling and querying Greek legislation using semantic web technologies. In *The Semantic Web (ESWC 2017)* (Lecture Notes in Computer Science). Vol. 10249. Springer, Cham, Switzerland, 591–606.

[5] Thiago Raulino Dal Pont, Federico Galli, Galileo Sartor, and Giuseppe Contissa. 2025. "lost in EU regulation? Don't worry, AI found the obligation" — extracting and representing legal obligations in the GDPR, the DSA, and the AI Act. In *Proc. 20th Int. Conf. Artificial Intelligence and Law (ICAIL)*, 129–138. DOI: 10.1145/3769126.3769260.

[6] Harshil Darji, Martin Heckelmann, Christina Kratsch, and Gerard de Melo. 2026. Bundesrecht: an open library and corpus for German statutory reference processing. arXiv:2605.31338. Preprint. (2026).

[7] Harshil Darji, Jelena Mitrović, and Michael Granitzer. 2023. A dataset of German legal reference annotations. In *Proc. 19th Int. Conf. Artificial Intelligence and Law (ICAIL)*, 392–396. DOI: 10.1145/3594536.3595173.

[8] Emile de Maat, Radboud Winkels, and Tom van Engers. 2006. Automated detection of reference structures in law. In *Legal Knowledge and Information Systems (JURIX 2006)* (Frontiers in Artificial Intelligence and Applications). Vol. 152. IOS Press, Amsterdam, The Netherlands, 41–50.

[9] European Commission. 2021. Ref2Link. ISA^2 LegIT action 2016.38 presentation. [Online]. Available: https://interoperable-europe.ec.europa.eu/sites/default/files/document/2021-06/Ref2link_pres_1_7_short.pdf. Accessed: Aug. 27, 2026. (June 2021).

[10] European Commission Legal Service. 2025. Ref2Link. Kit v1.3.32, EUPL-1.2. [Online]. Available: https://interoperable-europe.ec.europa.eu/collection/justice-law-and-security/solution/ref2link. Accessed: Aug. 27, 2026. (Oct. 2025).

[11] European Commission, Legal Service. 2015. *Joint Practical Guide of the European Parliament, the Council and the Commission for Persons Involved in the Drafting of European Union Legislation*. Catalogue no. KB-02-13-228-EN-N. Print edition: ISBN 978-92-79-49121-4, doi 10.2880/89965. Publications Office of the European Union, Luxembourg. ISBN: 978-92-79-49084-2. DOI: 10.2880/5575.

[12] Nicola Fabiano. 2026. Qualified cross-references as a verification method: the normative environment of the EU AI Act. arXiv:2608.19194. Preprint. (Aug. 2026).

[13] Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. In *Proc. Natural Legal Language Processing Workshop (NLLP) at EMNLP*. Abu Dhabi, UAE, 184–193.

[14] Ioannis Karamitsos, Stefanos Roufas, Khalil Al-Hussaeni, and Andreas Kanavos. 2025. LegNER: a domain-adapted transformer for legal named entity recognition and text anonymization. *Frontiers in Artificial Intelligence*, 8, 1638971. DOI: 10.3389/frai.2025.1638971.

[15] Konrad Kollnig. 2026. LegalViz.EU. Law & Tech Lab, Maastricht University, GPL-3.0, `main` branch. [Online]. Available: https://github.com/maastrichtlawtech/legalviz.eu. Accessed: Aug. 28, 2026. (Aug. 2026).

[16] Marios Koniaris, Ioannis Anagnostopoulos, and Yannis Vassiliou. 2018. Network analysis in the legal domain: a complex model for European Union legal sources. *Journal of Complex Networks*, 6, 2, (Apr. 2018), 243–268.

[17] Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. 2020. A dataset of German legal documents for named entity recognition. In *Proc. 12th Language Resources and Evaluation Conf. (LREC)*. Marseille, France, 4478–4485.

[18] Lexemo. 2024. `Roberta_large_legal_act_extraction`. RoBERTa-large token classifier, MIT licence. [Online]. Available: https://huggingface.co/Lexemo/roberta_large_legal_act_extraction. Accessed: Aug. 27, 2026. (2024).

[19] Yinhan Liu et al. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

[20] Mark David Longar, Erik Novak, and Marko Grobelnik. 2024. Creating local world models using LLMs. In *Data Mining and Data Warehouses (SiKDD), Proc. 27th Int. Multiconference Information Society (IS 2024), Volume C*. Ljubljana, Slovenia, (Oct. 2024). DOI: 10.70314/is.2024.sikdd.22.

[21] Pedro H. Luz de Araujo, Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. 2018. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In *Computational Processing of the Portuguese Language (PROPOR 2018)* (Lecture Notes in Computer Science). Vol. 11122. Springer, Cham, Switzerland, 313–323. DOI: 10.1007/978-3-319-99722-3_32.

[22] Vasile Păiș, Maria Mitrofan, Carol Luca Gasan, Vlad Ianov, Corvin Ghiță, Sergiu Coneschi, and Andrei Onuț. 2024. LegalNERo: a linked corpus for named entity recognition in the Romanian legal domain. *Semantic Web*, 15, 3, 831–844.

[23] Publications Office of the European Union. [n. d.] ELI for subdivisions: specifications for the identification of subdivisions in EU legislation using ELI. v2. [Online]. Available: https://eur-lex.europa.eu/content/eli-register/ELI-subdivisions-specifications-v2.pdf. Accessed: Aug. 27, 2026. ().

[24] Publications Office of the European Union. 2026. Interinstitutional style guide. [Online]. Available: https://style-guide.europa.eu/en/. Accessed: Aug. 31, 2026. Continuously revised; the site carries no version or revision date, so the access date is the only pin. (2026).

[25] Theresia Veronika Rampisela, Henrik Palmer Olsen, and Giovanni Colavizza. 2026. LegalPincite: multi-level legal information retrieval dataset. arXiv:2608.03756. Preprint. (Aug. 2026). DOI: 10.48550/arXiv.2608.03756.

[26] Nicolas Sannier, Morayo Adedjouma, Mehrdad Sabetzadeh, and Lionel C. Briand. 2017. An automated framework for detection and resolution of cross references in legal texts. *Requirements Engineering*, 22, 2, 215–237. Online-first Nov. 2015; issue 2017. DOI: 10.1007/s00766-015-0241-3.

[27] Holli Sargeant, Robert Östling, and Måns Magnusson. 2025. Detecting legal citations in United Kingdom court judgments. In *Proc. 2025 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 26810–26836.

[28] Giovanni Sartor, Piera Santin, Davide Audrito, Emilio Sulis, and Luigi Di Caro. 2022. Automated extraction and representation of citation network: a CJEU case-study. In *Advances in Conceptual Modeling (ER 2022 Workshops)* (Lecture Notes in Computer Science). Vol. 13650. Springer, Cham, Switzerland, 102–111. DOI: 10.1007/978-3-031-22036-4_10.

[29] Robert Shaffer and Stephen Mayhew. 2019. Legal linking: citation resolution and suggestion in constitutional law. In *Proc. Natural Legal Language Processing Workshop (NLLP) at NAACL*. Minneapolis, MN, USA, 39–44.

[30] Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. 2014. Automated reference resolution in legal texts. *Artificial Intelligence and Law*, 22, 1, (Mar. 2014), 29–60. DOI: 10.1007/s10506-013-9149-8.

Towards Scientific Contradiction Mining: Topic Discovery and Claim Extraction from AI Literature

Andrej Natev
an49507@student.uni-lj.si
Faculty of Computer and Information Science, University of Ljubljana
Ljubljana, Slovenia

Elham Motamedi
elham.motamedi@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Scientific contradiction mining requires identifying papers that discuss comparable problems and recovering the empirical findings that may support or contradict one another. This work evaluates two prerequisites for such a pipeline. First, we investigate whether consensus across three scientific document encoders can identify stable and coherent topic groups. Second, we examine whether restricting extraction to explicitly result-oriented sections is sufficient or whether complete full-text processing is necessary. Experiments are conducted on 4,227 papers from the arXiv Computer Science – Artificial Intelligence category (`cs.AI`) using SciNCL, SPECTER, SPECTER2, and BERTopic. The strongest clustering configuration achieved a three-way Jaccard coefficient of 0.885, while manual evaluation of seven high-confidence groups produced 97.5% topic coherence. In a focused full-text case study, 1,147 automatically extracted candidates were manually evaluated, yielding 510 valid result-bearing findings. Result-oriented sections yielded substantially higher extraction precision than other sections (72.0% vs 34.1%) and a higher proportion of findings containing quantitative evidence (55.8% vs 37.3%). However, 55.7% of all validated findings occurred outside result-oriented sections, indicating a trade-off between precision and quantitative density on the one hand and broader recovery of validated findings on the other. These evaluations also produced two manually annotated reference datasets for topic coherence and scientific finding extraction. These results support a staged contradiction-mining pipeline in which cross-encoder consensus narrows the literature and full-text processing supplies evidence for subsequent claim decontextualization and contradiction detection.

Keywords

scientific literature mining, document embeddings, topic modeling, scientific claim extraction

1 Introduction

The increasing volume of scientific literature makes it difficult to determine how findings reported across different studies relate to one another. Papers addressing similar problems may report supporting, conflicting, or apparently contradictory results because of differences in datasets, models, experimental conditions, or evaluation procedures. Automatically identifying such relationships could support literature review, evidence synthesis, and the discovery of unresolved research questions.

Scientific contradiction mining requires more than identifying documents that discuss similar topics. Relevant papers must first be grouped reliably, after which their empirical findings must be recovered and transformed into comparable scientific claims. Previous work on scientific claim verification emphasises atomicity (a claim should express a single proposition), faithfulness (it should preserve the meaning of the source text), and self-containment (it should remain interpretable without relying on surrounding context) as important properties of claim formulation, since claims with these properties are a prerequisite for reliable comparison and downstream contradiction detection [9, 12, 5]. Contradiction-detection studies further show that contextual differences may distinguish genuine contradictions from condition-dependent results [4, 3].

We therefore investigate two prerequisites for scientific contradiction mining:

- **RQ1:** Does cross-encoder consensus identify stable and coherent scientific topic groups?
- **RQ2:** To what extent does full-text processing recover result-bearing findings beyond explicitly result-oriented sections?

RQ1 is evaluated on a corpus of 4,227 papers from the arXiv Computer Science – Artificial Intelligence category (`cs.AI`), while RQ2 is examined through a focused full-text case study within one high-confidence consensus group. The complete intended pipeline is illustrated in Figure 1. The present work evaluates topic modelling and scientific finding extraction, while reliable atomic claim decontextualization and contradiction detection are left for future work.

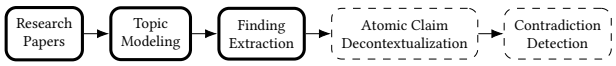


Figure 1: Intended scientific contradiction-mining pipeline. Solid stages are investigated in this work, while dashed stages are left for future work.

2 Related Work

Scientific document encoders such as SPECTER [1], SciNCL [6], and SPECTER2 [8] provide semantic representations for organising scientific literature, while BERTopic combines embeddings, dimensionality reduction, density-based clustering, and class-based TF–IDF to construct interpretable topic groups [2]. We use these representations to examine agreement across separately constructed encoder-specific topic spaces.

Scientific claim-verification systems generally assume that a claim is already available. SciFact introduced expert-written claims paired with supporting or refuting evidence [9], while later

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.119

work studied automatic claim generation and refinement [12, 5] and full-document or open-domain verification [11, 10]. Contradiction-focused studies further show that apparent conflicts may depend on contextual and experimental conditions [4, 3].

Our work addresses an earlier stage of this pipeline: rather than starting from predefined claims or candidate statement pairs, we first narrow a broad scientific corpus through cross-encoder topic consensus and then recover result-bearing findings from complete paper text for subsequent decontextualization and contradiction analysis.

3 Methods

This section describes the data and methodology used to address the two research questions. We first present the construction of the scientific paper corpus, followed by the cross-encoder topic modelling and consensus procedure linked to RQ1 and the full-text scientific finding extraction and validation procedure linked to RQ2.

3.1 Corpus

We used the structured UnarXive collection [7]. From an initial computer-science subset of 226,808 papers, we selected publications whose category was exactly `cs.AI`, retained English documents with cleaned abstracts of at least 200 characters, and removed duplicate paper–abstract combinations. The final corpus contained 4,227 papers. Titles and abstracts were used for topic discovery, while structured full-text sections were retained for subsequent finding extraction.

3.2 Topic Modelling and Cross-Encoder Consensus

Each paper was separately represented using SciNCL [6], SPECTER [1], and SPECTER2 [8]. Preliminary experiments also considered several general-purpose and scientific representations, including MiniLM, MPNet, DistilBERT/DistilRoBERTa, and SciBERT; SciNCL, SPECTER, and SPECTER2 were retained because qualitative inspection during preliminary experiments indicated clearer and more interpretable topic structures for the subsequent consensus analysis. BERTopic was then applied separately to each representation using UMAP and HDBSCAN.

The HDBSCAN minimum cluster size specifies the smallest group that can be retained as a cluster, whereas minimum samples controls the conservativeness of the density criterion and therefore the tendency to classify documents as noise. Several parameter combinations were explored during preliminary experiments. Two configurations producing useful topic structures at different resolutions were retained: coarse clustering with a minimum cluster size of 15 and minimum samples of 3, and granular clustering with values of 10 and 2, respectively. These configurations were selected empirically during exploratory analysis rather than optimized against the subsequent manual coherence labels.

Because numerical topic identifiers are specific to each encoder, consensus groups were defined through shared paper membership rather than topic numbers. Paper-level topic assignments from SciNCL, SPECTER, and SPECTER2 were merged, and papers classified as outliers by any encoder were excluded. Each observed combination of three encoder-specific topic assignments was treated as a candidate consensus group. Cross-encoder agreement was quantified using the strict three-model intersection, three-way and pairwise Jaccard coefficients, and model retention, defined as the proportion of each encoder-specific topic preserved in the strict three-model intersection.

We evaluated both the original HDBSCAN assignments (*native*) and assignments obtained after BERTopic outlier reassignment (*reduced*), producing four configurations: coarse/native, coarse/reduced, granular/native, and granular/reduced.

Consensus groups were required to contain at least 10 papers in their strict intersection. Eligible groups were ranked by three-way Jaccard coefficient, followed by minimum model retention and intersection size. Seven high-confidence granular/native groups additionally satisfied a three-way Jaccard coefficient of at least 0.50 and minimum model retention of at least 0.50. Papers from the union of the corresponding encoder topics were manually rated as clearly relevant (2), partially relevant (1), or unrelated (0). This produced a manually annotated reference dataset of 280 paper–topic judgments for evaluating topic coherence. Topic coherence was calculated as the proportion of paper–topic judgments rated clearly relevant (2). For comparison with individual encoder-specific topics, the same annotations were restricted to the papers belonging to each encoder-specific topic and to the strict three-encoder intersection. Coherence was calculated separately for SciNCL, SPECTER, SPECTER2, and the strict consensus, and macro-averaged across the seven evaluated groups.

3.3 Scientific Finding Extraction and Validation

For RQ2, we selected the 21-paper strict intersection of the backdoor and adversarial-attack consensus group as a focused case study. The group exhibited strong cross-encoder agreement (J_3 = 0.778) and an experimental focus suitable for evaluating result-bearing finding extraction. Structured full text was available for 20 papers, all of which were processed by a sequential full-text reader based on the 4-bit quantized Qwen2.5-7B-Instruct model. Each paper was divided into text chunks that were processed in document order. From each chunk, the model extracted candidate result-bearing findings together with their source section and supporting textual evidence.

We consider experimental outcomes, comparisons, observed effects, robustness results, ablations, sensitivity analyses, transferability, generalisation, and failure cases to be result-bearing findings. Method descriptions, experimental setup, dataset descriptions, metric definitions, research questions, background information, and future work are excluded.

All 1,147 extracted candidates were manually classified as either valid result-bearing findings or non-findings. This produced a second manually annotated reference dataset for evaluating scientific finding extraction.

For section-level analysis, headings were classified using a keyword-based rule. A section was considered *result-oriented* when its heading indicated empirical evaluation, including conventional labels such as Results, Experiments, and Evaluation and specialised terms referring to performance, comparison, ablation, effectiveness, robustness, transferability, and empirical analysis. All remaining headings were classified as other full-text sections. Extraction restricted to result-oriented sections was treated as a section-selection baseline against which complete full-text processing was compared.

We additionally identified explicit quantitative evidence using a conservative pattern-based rule covering percentages, scores

and decimal values, rates, counts, and measured values with units. Bare numeric identifiers, such as numbers occurring only in model or dataset names, were excluded.

4 Results

This section presents the empirical results for both research questions. We first evaluate the stability and semantic coherence of the cross-encoder topic groups for RQ1, and then examine extraction precision and the distribution of validated and quantitative findings across result-oriented and other full-text sections for RQ2.

Table 1: Cross-encoder consensus across clustering configurations. Best J_3 denotes the highest three-way Jaccard coefficient, and $R_{\min}$ is the minimum model retention of that same top-ranked consensus group. Mean top-10 J_3 is the mean three-way Jaccard coefficient across the ten highest-ranked eligible groups.

Configuration	Groups	Best J_3	$R_{\min}$	Mean top-10 J_3
Coarse/native	53	0.846	0.880	0.530
Coarse/reduced	83	0.636	0.695	0.351
Granular/native	38	**0.885**	**0.896**	**0.613**
Granular/reduced	68	0.601	0.664	0.391

4.1 Topic Stability and Coherence

Table 1 summarises cross-encoder agreement for the four clustering configurations. The granular/native configuration produced the strongest high-confidence consensus, achieving a best three-way Jaccard coefficient of 0.885, minimum model retention of 0.896, and a mean top-10 Jaccard of 0.613.

Outlier reassignment increased document coverage but consistently reduced cross-encoder agreement. For coarse clustering, mean top-10 three-way Jaccard decreased from 0.530 to 0.351 after reassignment, while for granular clustering it decreased from 0.613 to 0.391. This indicates a trade-off between assigning more documents to topics and retaining high-confidence agreement across different encoder representations.

Seven granular/native groups satisfied the high-confidence criteria. They represented argumentation, backdoor and adversarial attacks, temporal reasoning, graph learning, process mining, AI safety, and multi-agent reinforcement learning.

Human evaluation produced a reference dataset of 280 paper–topic judgments across the unions of the seven corresponding encoder topics. Of these, 273 were rated clearly relevant and seven unrelated, corresponding to an overall union-level coherence of **97.5%**. No papers received the intermediate score. Union-level coherence ranged from 92.3% to 100% across individual groups.

Using the same annotations for the individual-encoder comparison, macro-average coherence was 98.1% for SciNCL, 98.6% for SPECTER, and 98.0% for SPECTER2, compared with 99.0% for the strict three-encoder consensus. The mean individual-encoder coherence was 98.2%, so consensus provided a modest improvement of approximately 0.7 percentage points.

Several topic groups were also robust to changes in clustering configuration. The 21-paper backdoor/adversarial consensus core was identical under coarse/native and granular/native clustering ($J = 1.0$). The process-mining group achieved $J = 0.964$ between coarse/native and granular/native and $J = 1.0$ between granular/native and granular/reduced. The highest-ranked argumentation group achieved $J = 0.957$ between coarse/native and granular/native.

Table 2: High-confidence granular/native consensus groups. $|I|$ denotes the three-encoder intersection size, J_3 the three-way Jaccard coefficient, $R_{\min}$ minimum model retention, and Coh. manually evaluated coherence over the union of the corresponding encoder topics.

Topic	$\|I\|$	J_3	$R_{\min}$	Coh.
Argumentation	69	0.885	0.896	96.2%
Backdoor attacks	21	0.778	0.840	96.3%
Temporal reasoning	18	0.692	0.750	100%
Graph learning	17	0.654	0.773	92.3%
Process mining	27	0.628	0.711	97.7%
AI safety	14	0.538	0.636	100%
Multi-agent RL	29	0.537	0.644	100%

The backdoor/adversarial group consisted of encoder topics containing 23, 24, and 25 papers, with 21 papers in their strict intersection and 27 papers in their union, corresponding to a three-way Jaccard coefficient of **0.778**. Model retention was 0.913 for SciNCL, 0.875 for SPECTER, and 0.840 for SPECTER2. Together, these results indicate that agreement across separately generated encoder-specific topic structures can identify stable and highly coherent scientific neighbourhoods.

4.2 Full-Text Finding Extraction

The focused RQ2 case study processed 471 full-text chunks and generated 1,147 candidate result-bearing findings. All candidates were manually evaluated, yielding 510 valid findings and 637 non-findings, corresponding to an overall precision of **44.5%**.

Table 3: Extraction results by section type. Quantitative evidence percentages are calculated over valid findings within each section type.

Section type	Candidates	Valid	Non-valid	Quant. evidence
Result-oriented	314	226 (72.0%)	88 (28.0%)	55.8%
Other full text	833	284 (34.1%)	549 (65.9%)	37.3%
All	1,147	510 (44.5%)	637 (55.5%)	45.5%

Of the 1,147 extracted candidates, 314 originated from result-oriented sections and 833 from other full-text sections. Result-oriented sections contained 226 valid findings and 88 non-findings, corresponding to a substantially higher precision of **72.0%**. In comparison, 284 of the 833 candidates extracted from other full-text sections were valid, corresponding to a precision of **34.1%**.

Despite their lower precision, other full-text sections contributed 284 of the 510 validated findings, or **55.7%** of all valid findings. Result-oriented sections accounted for the remaining 226 findings (**44.3%**). Thus, restricting extraction to explicitly result-oriented sections would improve precision but would exclude more than half of the validated findings identified through full-text processing.

Quantitative-evidence analysis further showed that **55.8%** of valid findings from result-oriented sections contained explicit quantitative evidence. Importantly, **37.3%** of valid findings from other full-text sections also contained quantitative evidence. Across all 510 valid findings, 45.5% contained an explicit reported

quantity. This indicates that findings occurring outside result-oriented sections are not limited to qualitative discussion and frequently contain measurable empirical evidence relevant to downstream scientific comparison.

Taken together, these results reveal a clear precision–breadth trade-off: result-oriented sections provide a substantially denser source of valid and quantitatively supported findings, while processing the complete document recovers a larger overall set of validated findings.

5 Discussion and Conclusion

The results support a staged approach to scientific contradiction mining. For RQ1, consensus across three scientific document encoders produced highly coherent topic groups, several of which remained stable under alternative clustering configurations. Native HDBSCAN assignments consistently achieved stronger cross-encoder agreement than outlier-reassigned topics, indicating a trade-off between document coverage and topic confidence. Manual evaluation additionally produced a 280-judgment reference dataset with an overall topic coherence of 97.5%.

The RQ2 case study reveals a clear trade-off between precision, quantitative evidence, and breadth of finding recovery. Result-oriented sections were substantially more precise than other full-text sections (72.0% vs 34.1%) and contained a higher proportion of quantitatively supported findings (55.8% vs 37.3%). However, they accounted for only 44.3% of all validated findings, meaning that restricting extraction to these sections would exclude 55.7% of the findings recovered through full-text processing. Within this case study, result-oriented sections therefore provide a higher-precision and more quantitatively dense source of evidence, whereas full-text processing recovers a larger set of validated findings.

The overall extraction precision of 44.5% nevertheless shows that identifying result-bearing scientific findings from unrestricted full text remains challenging. Additional filtering and refinement are therefore required before extracted findings can be used reliably for downstream contradiction detection. The present evaluation measures precision among generated candidates but does not estimate extraction recall. Because the source papers were not exhaustively annotated for all result-bearing findings separately from the reader-generated candidates, findings that were never proposed as candidates may remain unobserved. Accordingly, the reported section-level differences describe the distribution of validated extracted findings rather than exhaustive recall over all findings present in the source papers.

A further limitation is that the manual evaluations were produced by a single annotator, so inter-annotator agreement could not be estimated. Future evaluation should therefore involve multiple independent annotators to quantify annotation reliability. Another challenge is transforming findings into atomic and self-contained claims while preserving entities, experimental conditions, comparisons, and quantitative values. Preliminary experiments showed that automatic decontextualization can alter semantic roles or omit important conditions; contradiction classification was therefore not evaluated in the present study.

The two manually annotated datasets created to address RQ1 and RQ2 are publicly available on Zenodo[1] and provide reference sets for evaluating future improvements to topic selection and scientific finding extraction. Overall, the results suggest that cross-encoder consensus can narrow a broad scientific corpus to coherent neighbourhoods, while the RQ2 case study indicates that full-text processing can recover empirical evidence that would be missed by approaches restricted to explicitly result-oriented sections.

Future work will extend the pipeline beyond the two stages evaluated here. First, finding extraction can be improved using the manually annotated RQ2 dataset to reduce false positives while preserving the additional validated findings recovered through full-text processing. Second, extracted findings should be transformed into atomic, self-contained claims through decontextualization methods that explicitly preserve experimental conditions and quantitative evidence. Third, comparable claims can be retrieved within and across the consensus topic groups before applying condition-aware contradiction classification capable of distinguishing genuine disagreement from results obtained under different settings. Finally, the full-text case study should be replicated across additional consensus groups and scientific domains, and evaluation should include multiple independent annotators to assess annotation agreement and the generalisability of the proposed pipeline.

[1] doi:10.5281/zenodo.22968108

Acknowledgements

This research was funded through the projects ELIAS (HE 101120237), DataPACT (HE 101189771), ARIS Gravitacija (GC-0001).

References

[1] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: document-level representation learning using citation-informed transformers. In *Proceedings of ACL 2020*. Association for Computational Linguistics, 2270–2282. doi:10.18653/v1/2020.acl-main.207.

[2] Maarten Grootendorst. 2022. BERTopic: neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

[3] Ashwin Kirubakaran and Henry Gagnier. 2026. BioConflict: a benchmark for evaluating large language models in biomedical contradiction detection and consensus synthesis. In *Proceedings of BioNLP 2026*. Association for Computational Linguistics, 552–558. doi:10.18653/v1/2026.bionlp-1.44.

[4] Dave Makhervaks, Plia Gillis, and Kira Radinsky. 2023. Clinical contradiction detection. In *Proceedings of EMNLP 2023*. Association for Computational Linguistics, 1248–1263. doi:10.18653/v1/2023.emnlp-main.80.

[5] Raúl Ortega and Jose Manuel Gomez-Perez. 2025. SciClaims: an end-to-end generative system for biomedical claim analysis. In *Proceedings of EMNLP 2025: System Demonstrations*. Association for Computational Linguistics, 141–154. doi:10.18653/v1/2025.emnlp-demos.11.

[6] Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of EMNLP 2022*. Association for Computational Linguistics, 11670–11688. doi:10.18653/v1/2022.emnlp-main.802.

[7] Tarek Saier, Johan Krause, and Michael Färber. 2023. unarXive 2022: all arxiv publications pre-processed for NLP, including structured full-text and citation network. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL 2023)*. IEEE, 66–70. doi:10.1109/JCDL57899.2023.00020.

[8] Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: a multi-format benchmark for scientific document representations. In *Proceedings of EMNLP 2023*. Association for Computational Linguistics, 5548–5566. doi:10.18653/v1/2023.emnlp-main.338.

[9] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: verifying scientific claims. In *Proceedings of EMNLP 2020*. Association for Computational Linguistics, 7534–7550. doi:10.18653/v1/2020.emnlp-main.609.

[10] David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. SciFact-Open: towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, 4719–4734. doi:10.18653/v1/2022.findings-emnlp.347.

[11] David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. MultiVerS: improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, 61–76. doi:10.18653/v1/2022.findings-naacl.6.

[12] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of ACL 2022 (Volume 1: Long Papers)*. Association for Computational Linguistics, 2448–2460. doi:10.18653/v1/2022.acl-long.175.

Chronos-2 for Spacecraft Telemetry Forecasting

Žiga Klun
Bias Variance Labs d.o.o.
Ljubljana, Slovenia
zk56034@student.uni-lj.si

Dragi Kocev
Bias Variance Labs d.o.o.
Jožef Stefan Institute
Ljubljana, Slovenia

Jure Brence
Bias Variance Labs d.o.o.
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Spacecraft telemetry is continuously monitored for anomalies, and the dominant automated approach forecasts nominal behaviour and applies a threshold to the resulting residuals, so the detection quality is affected by the forecast quality. We compare the per-channel LSTM inspired by Telemanom against Chronos-2, a pretrained model supporting multivariate and covariate-informed forecasting, on 76 channels of the ESA Anomaly Dataset. The best results come from Chronos-2 fine-tuned with LoRA, followed closely by the same model applied in zero-shot mode. In both cases, Chronos-2 reduces the median normalised mean squared error by over 50% compared to the strongest LSTM baseline. The small improvement from fine-tuning indicates that most of Chronos-2's advantage is available without mission-specific adaptation. Supplying telecommands as covariates does not meaningfully affect the performance of Chronos-2, which we attribute to the exclusion of rare events from our evaluation windows.

Keywords

spacecraft telemetry, time series forecasting, foundation models, zero-shot forecasting, anomaly detection, space operations

1 Introduction

Spacecraft continuously downlink telemetry in the form of time series that record temperatures, voltages, currents, attitude, and the state of on-board instruments. A single satellite can produce recordings from thousands of such channels, and monitoring them is a daily responsibility of operations engineers in mission control centres [4, 7]. The consequences of missing a fault are severe, since a spacecraft cannot be repaired once launched and a problem that goes unnoticed can escalate into the loss of an instrument or of the entire mission. At the same time, the volume of downlinked data keeps growing, which makes manual inspection increasingly impractical.

Operational anomaly detection still relies mostly on out-of-limits checks combined with manual analysis [4, 7], which is blind to anomalies that stay inside the nominal limits but break the expected temporal or cross-channel pattern. The dominant data-driven alternative is *forecasting-based* detection, in which a model trained on nominal telemetry predicts how each channel should continue and flags an anomaly when the prediction error is high enough. Detection quality is therefore affected by forecast quality, since a forecaster that cannot represent nominal dynamics produces large residuals during nominal operation, and those become false alarms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.104

The canonical instance of this design is Telemanom [4], which trains one LSTM per channel and applies an unsupervised dynamic threshold to the resulting prediction errors. It remains the best performing algorithm on Mission 1 of the ESA Anomaly Detection Benchmark (ESA-ADB) [7], and is therefore the baseline we compare against. Two of its properties motivate this work. It forecasts one channel at a time, so cross-channel interactions are unused, and it produces a point prediction with no measure of its own uncertainty, so a separate stage is needed to decide which errors are large enough to matter.

Transformer-based foundation models, pretrained once and applied across forecasting and anomaly detection alike, have meanwhile become an important paradigm in time series analysis [8]. Chronos-2 [1], a recent model of this kind, addresses both limitations. It performs univariate, multivariate and covariate-informed forecasting zero-shot, predicts a long horizon in one pass, and emits quantiles. We therefore isolate the forecasting component, which can be evaluated without committing to any thresholding scheme, and compare an LSTM baseline against Chronos-2 on Mission 1 of ESA-ADB [7], under several input configurations, zero-shot and LoRA fine-tuned, and with telecommands as covariates.

2 Related Work

2.1 Anomaly detection in spacecraft telemetry

Expert systems, nearest neighbour searches and clustering have all been deployed operationally, yet out-of-limits checking remained the most widely deployed approach because of its low cost and interpretability [4]. Deep learning shifted the field towards forecasting-based detection, in which a network trained on nominal data acts as a predictor and its prediction errors are scored for anomalousness [12]. Its merits against classification-based alternatives have since been measured directly [10]. Telemanom [4] pairs per-channel LSTM forecasting with unsupervised dynamic thresholding of the prediction errors. Benecki et al. [2] showed that this thresholding stage is itself highly tunable, and Lakey and Schlippe compared recurrent, convolutional and transformer architectures for the task [9]. Beyond forecasting-based methods, reconstruction-based anomaly detection uses autoencoders to compress nominal telemetry and score the reconstruction error [6, 16]. ESA-ADB benchmarks one such model, DC-VAE-ESA, but forecasting-based Telemanom remains the strongest performer on Mission 1 [7]. Progress was long limited by a lack of realistic labelled data. The SMAP and MSL sets released with Telemanom [4] were the de facto standard for years, until Wu and Keogh [15] showed that these and other popular benchmarks suffer from flaws severe enough to make much of the reported progress illusory. OPS-SAT [13] added annotated telemetry from an ESA CubeSat, and ESA-ADB [7] is the most recent and by far the largest, with orders of magnitude more data points than any earlier public satellite telemetry dataset.

2.2 Time series forecasting

Classical methods such as ARIMA and exponential smoothing are *local*, fitting one set of parameters per series [5], while deep learning introduced *global* models sharing parameters across series, including DeepAR [14], which produces probabilistic rather than point forecasts, and the Temporal Fusion Transformer [11], which handles known-future and past-only covariates. Pretrained models form a third paradigm, trained once and applied to unseen data without task-specific training [1]. Most are univariate and ignore covariates, which limits their use on telemetry, where channels are coupled and contextual data, such as telecommands, can carry critical information.

3 Experimental Setup

3.1 The ESA Anomaly Benchmark

ESA-ADB is a benchmark for anomaly detection in satellite telemetry data, assembled by ESA operations engineers together with machine learning researchers from real, manually annotated telemetry of three ESA missions [7]. We use Mission 1 of ESA-ADB, the longest-running of its three missions. It holds 76 telemetry channels recorded over an anonymised span of about 14 years, some 775 million points, annotated by operations engineers with 200 events of which 118 are anomalies and 78 are rare nominal events such as commanded manoeuvres. The data is kept close to what operators actually see, including irregular timestamps, varying sampling rates and communication gaps.

3.2 Forecasting task construction

ESA-ADB defines an anomaly detection task. As our object of study is the forecaster, we derive a forecasting task from the same data so that every model sees identical inputs and is scored on identical targets.

All 76 channels are resampled to a common 1-minute grid to address the irregular timestamps of the raw telemetry. Observations within each one-minute bin are averaged and empty bins linearly interpolated without a maximum-gap cutoff. From this grid we cut fixed windows of $C = 2,048$ context steps followed by $H = 256$ horizon steps, C matching Chronos-2's pretraining context length, so a model observes roughly 34 hours of history and predicts the next 4.3 hours, with candidate start times advanced in steps of 3 hours. A window is accepted only if neither its context nor its horizon overlaps an annotated event on any of the 76 channels, irrespective of the model inputs, so that neither anomalies nor rare nominal events leak into inputs or targets. This yields 20,000 windows, split chronologically into 16,000 training, 2,000 validation and 2,000 test, which keeps future information out of training. At each split boundary, 13 windows were purged, separating retained origins by 2,520 minutes and preventing cross-split overlap.

Interpolated horizon values are scored, while each context is independently Gaussian-filtered. Each channel is then standardised using a fixed mean and standard deviation computed over the training period. Figure 1 shows one such window, with the context the model receives and the horizon it must predict. Per-window standardisation is avoided because, on near-constant channels, local variation can be far smaller than overall variation and thereby inflate physically negligible errors.

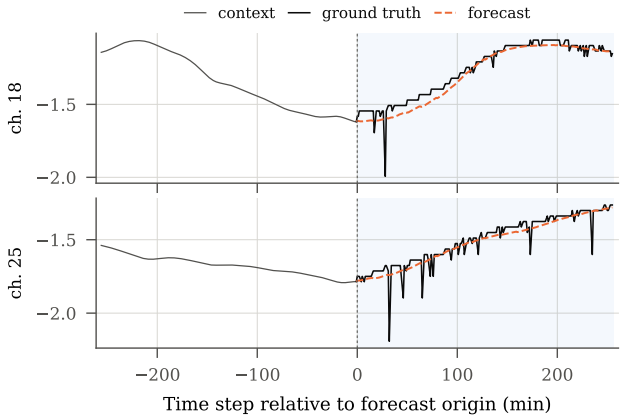


Figure 1: One test window forecast jointly by Chronos-2 (LoRA, multivariate), two of the 76 channels shown. The model sees only the context left of the dashed line. The context is Gaussian-filtered, the horizon is not. Only the final 256 of the 2048 context steps are shown.

3.3 Telecommands

The dataset also records the 698 telecommands sent to the spacecraft. These are natural explanatory inputs, since many sharp telemetry transitions are the intended consequence of a command rather than spontaneous behaviour. Following the ESA-ADB convention for full-channel experiments we keep only the commands of the highest priority. Each is a sparse event log, encoded on the telemetry grid as a binary indicator that is one in any interval containing an execution. These columns are passed to the models unchanged, without standardisation or smoothing.

4 Models

In our experiments, we compare two forecasting models.

- The **LSTM** is a Telemanom-inspired forecasting baseline [4]. It retains the two-layer recurrent architecture but replaces Telemanom's output head with a linear head that emits the complete horizon directly. We evaluate the original per-channel configuration together with fixed-capacity multivariable and multivariate variants; these are controlled configuration comparisons, not independently optimised multivariate LSTMs.
- **Chronos-2** [1] is a pretrained transformer that performs univariate, multivariate and covariate-informed forecasting in context, without task-specific training, and can forecast all 76 channels jointly. It distinguishes past-only covariates, whose values stop at the forecast origin, from known-future covariates, whose values extend across the horizon. Telecommands are known-future, since the command schedule is planned in advance. It is also probabilistic, predicting 21 quantiles from 0.01 to 0.99, so a predictive interval is supplied directly rather than reconstructed from smoothed residuals [2]. As the LSTM is a point forecaster, we use only the median quantile. We evaluate Chronos-2 zero-shot and fine-tuned with LoRA [3], which freezes the pretrained weights and injects trainable low-rank matrices.

4.1 Input configurations

The models are compared under three input configurations. In the *single-channel* configuration (SC) a model sees and predicts one channel only, reproducing Telemanom's original design. The *multivariate* configuration (MVT) gives one model all 76 channels as input and requires it to forecast all 76 jointly, so cross-channel structure is available for every target. The *multivariable* configuration (MVB) also takes all 76 channels as input but forecasts a single target channel, separating the benefit of extra context from the burden of predicting many outputs at once. Telecommands are added as extra input-only covariates on top of the multivariate configuration, either past-only or known-future, and are never forecast themselves.

5 Evaluation

The LSTM uses two layers with 64 hidden units, close to Telemanom's two-layer, 80-unit architecture, and is trained with Adam at a learning rate of 10^{-3}, a batch size of 64 and early stopping on the validation split. We hold this capacity fixed across configurations to isolate the effect of input and output dimensionality; because it was not tuned for 76 outputs, the MVT result may partly reflect a capacity bottleneck. Chronos-2 is used in its base 120M-parameter form, either zero-shot or adapted with LoRA of rank 16 and $\alpha = 32$ at a learning rate of 10^{-5} for 2000 steps. All runs were executed on a single NVIDIA B200 GPU.

We report mean squared error, mean absolute error and the coefficient of determination R^2, all in the normalised space. Each metric is computed per channel over the test windows and then aggregated across the 76 channels. We report both the mean and the median across channels, but treat the median as the headline statistic, because a handful of near-constant channels carry very large normalised errors and dominate any average. The gap between the two is itself informative, and we examine it in Section 7. We also track how error accumulates over the horizon. For each channel the squared error is averaged over test windows at each of the 256 horizon steps and then accumulated as a running mean over steps $1..k$, so the quantity at k is the mean error of a k-step forecast. These per-channel curves are aggregated by their median, and at $k = 256$ the value reproduces the horizon-averaged MSE.

6 Results

Table 1 reports every configuration over all 76 channels and Figure 2 the corresponding per-channel spread.

Chronos-2 has lower MSE and MAE than the evaluated LSTM baselines. Its best setting reaches a median MSE of 0.0021 against 0.0059 for the strongest LSTM baseline, a factor of 2.8, and the gap in median R^2 is 0.996 against 0.984. The more striking comparison is the zero-shot row: without task-specific training on this mission, Chronos-2 attains a median MSE of 0.0025, 2.3 times lower than an LSTM trained on it directly. Fine-tuning helps, but only modestly, improving median MSE from 0.0025 to 0.0021 and mean R^2 from 0.745 to 0.759. Almost all of the benefit is therefore already present in the pretrained model.

For the evaluated 64-unit LSTM, moving from separately trained single-channel models to one joint 76-output model raises median MSE from 0.0059 to 0.1968. This shows that the fixed-capacity baseline does not scale successfully to the joint-output setting, but should not be read as a general limitation of tuned multivariate LSTMs. Chronos-2 shows no analogous penalty in the configurations tested and obtains its lowest error while forecasting all 76 channels jointly.

Table 1: All configurations over the 76 channels, horizon-averaged, in normalised units.

	MSE		MAE		R^2	
Configuration	Mean	Median	Mean	Median	Mean	Median
Chronos-2 zero-shot, MVT	0.103	0.0025	0.121	0.0309	0.745	0.995
Chronos-2 LoRA, MVT	0.100	0.0021	0.114	0.0259	0.759	0.996
+ telecommands, past-only	**0.100**	0.0021	**0.113**	**0.0257**	**0.760**	**0.996**
+ telecommands, known-future	0.100	**0.0021**	0.113	0.0258	0.759	0.996
LSTM SC	0.482	0.0059	0.289	0.0515	0.759	0.984
LSTM MVT	1.263	0.1968	0.648	0.3819	0.210	0.592
+ telecommands	1.448	0.1666	0.626	0.3521	−0.592	0.670

Table 2: Horizon-averaged MSE ($\times 10^{-3}$, normalised) on the four restricted-target channels.

		LSTM			LoRA Chronos-2		
Channel	STD	MVB	MVT	SC	MVB	MVT	SC
69	0.0005	0.280	1.42	0.251	**0.246**	0.247	0.264
26	0.0396	49.9	532.8	15.9	**2.79**	2.85	3.83
14	0.0595	132.7	196.4	1.11	**1.01**	1.01	1.01
61	0.3540	6.93	370.5	0.0132	0.0122	0.0124	**0.0094**
Mean		47.4	275.3	4.31	**1.01**	1.03	1.28

The cumulative horizon-averaged error separates the models further (Figure 3). From $k = 1$ to $k = 256$, the running-mean MSE over steps $1, \ldots, k$ increases by a factor of 1.2 for Chronos-2 LoRA, 1.5 for the zero-shot model and 2.2 for LSTM SC.

Telecommands do not meaningfully affect the performance of Chronos-2. Supplying the highest-priority commands as covariates produces only very small differences in Table 1, and this holds whether they are given as past-only or as known-future covariates, even though the latter reveals the exact future command schedule. For the LSTM the effect is mixed: the median R^2 improves from 0.592 to 0.670, while the mean R^2 falls from 0.210 to −0.592 because the channels it already handled worst degrade further.

Table 2 compares all three input configurations on a small set of four channels rather than all 76, since the multivariable and single-channel settings each require a separately trained model per target. The four channels are selected to represent different standard deviations (STD), with `channel_69` having one of the smallest and `channel_61` one of the largest. Restricting the LSTM from 76 outputs to one improves its mean MSE from 275.3 to 47.4×10^{-3}, and narrowing the input improves it further to 4.31×10^{-3}. The same output restriction leaves Chronos-2 unchanged at 1.01×10^{-3}, and its single-channel result (1.28×10^{-3}) is worse. Even at its best the LSTM remains roughly 4.2 times behind.

7 Discussion

Fine-tuning helps, but the pretrained model already carries most of the benefit. In this temporal split, LoRA reduces median MSE by 17.6%, a modest observed gain whose variability was not assessed, whereas the zero-shot model, which has never seen this mission, already outperforms an LSTM trained directly on it. Operationally this matters more than the ranking itself. Telemanom requires one trained model per channel, and those models must be maintained as the spacecraft ages and its behaviour drifts, whereas a pretrained model can be applied to a new channel, or

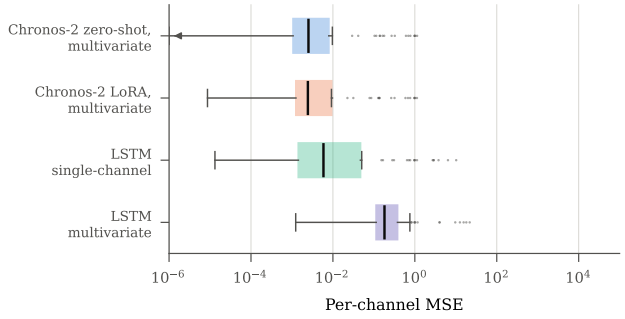


Figure 2: Per-channel horizon-averaged MSE, one box per configuration. The axis is truncated at 10^{-6}; the arrow indicates eight zero-shot values below the display floor, the smallest approximately 2×10^{-15}.

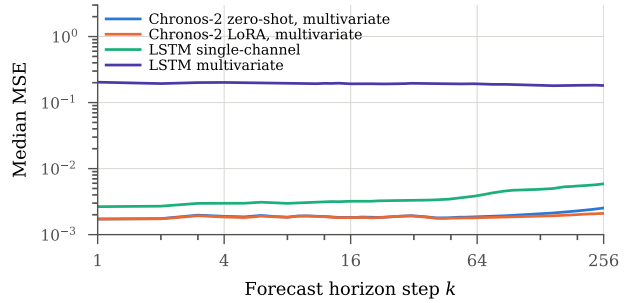


Figure 3: Median across the 76 channels of cumulative MSE over forecast steps $1, \ldots, k$.

an entirely new mission, with no training at all. Fine-tuning then becomes an optional refinement rather than a precondition for deployment.

Telecommands did not materially affect Chronos-2 (Table 1), even when revealed over the forecast horizon. ESA-ADB identifies commanded manoeuvres, resets and calibrations as rare nominal events and distinguishes commanded from anomalous resets [7]. Because our windows exclude every annotated event (Section 3), the transitions telecommands could explain are absent from training and test data.

The one-to-two-order-of-magnitude gap between mean and median MSE shows that a few difficult channels dominate the mean. We compute R^2 per channel relative to the training-mean predictor and aggregate across channels, omitting it when the baseline MSE is at most 10^{-8}; on near-constant `channel_69`, raw-unit MSE remains of order 10^{-10} even when R^2 is near or below zero. We therefore headline the median while reporting both aggregates and per-channel errors.

Limitations. All LSTM configurations share a fixed hidden size (Section 5), so the multivariate result characterises a fixed-capacity baseline rather than LSTMs in general. We evaluate one mission, point forecasts and a single LSTM baseline, leaving probabilistic detection and broader forecasting architectures for future work.

8 Conclusion

We compared an LSTM baseline against the foundation model Chronos-2 for forecasting spacecraft telemetry, with downstream anomaly detection as the goal. Chronos-2 is the stronger model in every configuration tested. It forecasts all 76 channels jointly and benefits from their interactions, whereas the LSTM degrades badly enough to be pushed towards one model per channel. Chronos-2's cumulative horizon-averaged error also grows slowly across the full forecast horizon. It achieves this in both zero-shot and fine-tuned settings, so a forecaster could be deployed on a new mission immediately, without first accumulating enough mission data to train on. The natural next step is to use the quantiles Chronos-2 already produces as the nominal envelope of a detector, replacing the residual thresholding stage that current methods depend on. Furthermore, evaluating the ability of Chronos-2 to leverage telecommand data in forecasting rare commanded events is crucial for anomaly detection.

Acknowledgements

We acknowledge the support of the European Space Agency (ESA), through the activity *AiSTRA-Nova: Knowledge Graph Reasoning for AI-Enhanced Spacecraft Health Monitoring*.

References

[1] Abdul Fatir Ansari et al. 2025. Chronos-2: From Univariate to Universal Forecasting. arXiv:2510.15821 [cs.LG]. (Oct. 2025). doi:10.48550/arXiv.2510.15821.

[2] Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. 2021. Detecting anomalies in spacecraft telemetry using evolutionary thresholding and LSTMs. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1143–1150. doi:10.1145/3449726.3459411.

[3] Edward J. Hu et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]. (Oct. 2021). doi:10.48550/arXiv.2106.09685.

[4] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 387–395. doi:10.1145/3219819.3219845.

[5] Rob J. Hyndman and George Athanasopoulos. 2021. *Forecasting: Principles and Practice*. (3rd ed.). OTexts, Melbourne, Australia.

[6] Weihua Jin, Bo Sun, Zhidong Li, Shijie Zhang, and Zhonggui Chen. 2019. Detecting Anomalies of Satellite Power Subsystem via Stage-Training Denoising Autoencoders. *Sensors*, 19, 14, (July 22, 2019), 3216. doi:10.3390/s19143216.

[7] Krzysztof Kotowski et al. 2025. European Space Agency Benchmark for Anomaly Detection in Satellite Telemetry. arXiv:2406.17826 [cs.LG]. (Aug. 2025). doi:10.48550/arXiv.2406.17826.

[8] Siva Rama Krishna Kottapalli et al. 2025. Foundation Models for Time Series: A Survey. (Apr. 5, 2025). arXiv: 2504.04011 [cs.LG]. Retrieved Aug. 31, 2026 from http://arxiv.org/abs/2504.04011. Preprint.

[9] Daniel Lakey and Tim Schlippe. 2024. A Comparison of Deep Learning Architectures for Spacecraft Anomaly Detection. In *2024 IEEE Aerospace Conference*. (Mar. 2024), 1–11. doi:10.1109/AERO58975.2024.10521015.

[10] Daniel Lakey and Tim Schlippe. 2024. Anomaly Detection in Spacecraft Telemetry: Forecasting vs. Classification. In *2024 IEEE Space Computing Conference (SCC)*. IEEE, Mountain View, CA, USA, (July 2024), 17–27. doi:10.1109/SCC61854.2024.00009.

[11] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37, 4, (Oct. 2021), 1748–1764. doi:10.1016/j.ijforecast.2021.03.012.

[12] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. *European Symposium on Artificial Neural Networks (ESANN)*, 89–94.

[13] Bogdan Ruszczak, Krzysztof Kotowski, David Evans, and Jakub Nalepa. 2025. The OPS-SAT benchmark for detecting anomalies in satellite telemetry. *Scientific Data*, 12, 1, (Apr. 2025), 710. doi:10.1038/s41597-025-05035-3.

[14] David Salinas, Valentin Flunkert, and Jan Gasthaus. 2019. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. arXiv:1704.04110 [cs.AI]. (Feb. 2019). doi:10.48550/arXiv.1704.04110.

[15] Renjie Wu and Eamonn J. Keogh. 2023. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. *IEEE Transactions on Knowledge and Data Engineering*, 35, 3, (Mar. 2023), 2421–2429. doi:10.1109/TKDE.2021.3112126.

[16] Zhaoping Xu, Zhijun Cheng, and Bo Guo. 2023. A Multivariate Anomaly Detector for Satellite Telemetry Data Using Temporal Attention-Based LSTM Autoencoder. *IEEE Transactions on Instrumentation and Measurement*, 72, 1–13. doi:10.1109/TIM.2023.3296125.

Biomedical Relation Extraction: What Lies Between the Entities?

Nadiia Shpanko
nadya.shpanko@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Erik Calcina
erik.calcina@ijs.si
Jožef Stefan International
Postgraduate School
Jožef Stefan Institute
Ljubljana, Slovenia

Erik Novak
erik.novak@ijs.si
Jožef Stefan International
Postgraduate School
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Clinical trial eligibility criteria are typically written in free text and contain medical concepts, numerical constraints, temporal conditions, and logical relations. Converting these criteria into structured representations requires identifying not only the relevant concepts but also the relations between them. In this work, we study relation classification between annotated entity mentions in the Chia corpus of clinical trial eligibility criteria. We compare three compact GLiNER2 model variants under the same fine-tuning and evaluation protocol. The Large model achieves the best held-out test performance with a micro-F1 score of 0.7510, compared with 0.7260 for Base and 0.7039 for Multilingual. The results show that compact, locally deployable models can perform effective biomedical relation classification while avoiding dependence on external inference services.

Keywords

Relation extraction, Biomedical NLP, Clinical trial eligibility criteria, Information extraction, Local inference

1 Introduction

Clinical trial eligibility criteria determine whether a patient can participate in a study. They combine diagnoses, procedures, medications, measurements, demographic restrictions, temporal conditions, and Boolean logic, yet are usually written as free text. Computational use therefore requires transformation of natural-language criteria into structured representations that support cohort identification, patient-trial matching, and query generation. The Chia corpus [4] was created for this purpose and represents eligibility statements as annotated entities connected by semantic and logical relationships.

Relation Extraction (RE) determines how concepts mentioned in a text are connected. Named Entity Recognition (NER) can identify a measurement such as haemoglobin or a condition such as hypertension, but mentions alone do not capture the relationships expressed by the criterion. A numerical threshold must be linked to its measurement, a temporal expression to the event it constrains, and logical alternatives through `AND` or `OR`. RE therefore requires relation existence and type to be determined for the relevant argument pair while preserving direction where applicable.

Biomedical RE is challenging because realistic schemas mix directed and symmetric relations, heterogeneous labels, and many unrelated candidate pairs. Strong entity recognition does not necessarily translate into strong relation extraction [6], and zero- or few-shot transfer remains unreliable [5]. We therefore focus on relation-specific fine-tuning.

We study these questions on Chia [4], which contains eleven semantic and logical relation types. Its mixture of directed and symmetric relations, multi-argument logical groups, complex mentions, and numerous unrelated candidate pairs makes it a useful evaluation setting for pairwise RE.

A second practical consideration is deployment. Clinical text may contain protected health information, making local processing attractive for privacy-sensitive applications [3]. GLiNER2 provides a compact schema-conditioned framework that can run locally without relying on an external language-model API [8].

This study makes three contributions. First, we formulate RE as reference-anchored pair-conditioned classification, separating relation existence and typing from mention detection. Second, we compare Base, Large, and Multilingual compact schema-driven model variants under the same parameter-efficient fine-tuning and evaluation protocol, including a direct comparison of the Multilingual and English Base checkpoints on the English-only Chia benchmark. Third, we analyse results per relation type and show that frequency alone does not explain difficulty.

The remainder of the paper is structured as follows. Section 2 reviews related work. Section 3 describes the relation extraction methodology. Section 4 presents the dataset, the training configuration, and the evaluation protocol. Section 5 reports the results, Section 6 discusses the findings and limitations, and Section 7 concludes the paper and outlines future work.

2 Related Work

Modern RE benchmarks increasingly contain heterogeneous entity and relation categories rather than a single interaction type. BioRED shows that RE remains difficult even when entity extraction is comparatively strong [6]. A common supervised strategy is to represent candidate spans and classify their pairs. Ji et al. proposed a span-based joint entity and relation model with contextual and span-specific representations [2], while PL-Marker models subject-object span pairs with explicit markers [7]. We take the same pair-conditioned view: the central decision is whether a pair of mentions is related and, if so, under which label.

The models we compare come from the schema-conditioned line of work. GLiNER introduced label-conditioned span extraction with a bidirectional Transformer encoder, allowing arbitrary labels without autoregressive generation [9]. GLiNER2 extends this approach to several information extraction tasks, including entity recognition, text classification, and hierarchical structured extraction [8]. Both are compact enough for local deployment.

Two further findings shape our setup. Labrak et al. report that instruction-finetuned models handle some clinical natural

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.112

language processing (NLP) tasks well while relation extraction remains weak under zero- and few-shot prompting [5], motivating fine-tuning in our setup. Bassignana and Plank note that reported RE scores depend strongly on protocol, including whether mentions are supplied or predicted, how direction is handled, and whether exact mention boundaries are required [1]. We therefore state the candidate construction explicitly and evaluate relation classification independently of mention detection.

3 Methodology

3.1 Relation extraction formulation

The task is formulated as single-label pair classification. As a concrete example from Chia [4], consider the eligibility criterion "patient has [Condition: hypertension] ([Measurement: systolic blood pressure] > 135 mmHg or [Measurement: diastolic blood pressure] > 85 mmHg)." The criterion contains the clinical concept "hypertension", two blood-pressure measurements, their numerical threshold values, and a logical `OR` connecting the two measurement conditions. This example illustrates the type of structured information represented in Chia and the relations that must be recovered between annotated concepts. In our experiments, the entity mentions and their types are supplied by the reference annotations, and the model is evaluated on the relations between these annotated mentions rather than on entity detection itself. Within each input unit, a single eligibility criterion in Chia, we enumerate all ordered pairs of distinct mentions taken from the reference annotations. No type- or distance-based filtering is applied. Both arguments are marked in the input together with their reference entity types, for example `[ARG1|Measurement]` and `[ARG2|Value]`, and the model assigns each pair a relation type or the `NA` label when no relation applies. We call this construction reference-anchored. Fixing the mentions separates relation existence and typing from mention detection, so the reported score reflects the relation decision and not the entity recogniser.

Direction is retained for semantic relations. For the symmetric `AND` and `OR` relations, the reverse orientation of an annotated edge is excluded from the candidate set rather than labelled `NA`, so these relations are scored only in the annotated orientation.

A pair that carries more than one relation label is expanded into one instance per reference label, and the instances are scored independently. Because the model returns one label per pair, every multi-label pair contributes at least one false negative under this evaluation.

3.2 Distribution-aware training construction

Enumerating all entity pairs produces a pronounced class imbalance, with negative examples substantially outnumbering positive ones. The pairwise expansion of Chia's logical annotations also inflates `OR` relative to the other relation types. Training on this distribution can bias the model toward predicting `NA` and hinder learning of the underrepresented relations.

We therefore construct a tempered training subset that reduces the imbalance without making all relation types equally frequent. Let n_i denote the number of positive training instances of relation class i. Each class is assigned a sampling weight proportional to $n_i^{0.85}$. Because the exponent is smaller than one, differences between frequent and rare classes are compressed: frequent relations remain more common, but their dominance is reduced. The weights are normalised to a fixed budget of positive instances, sampled randomly within each class, and an equal number of `NA` candidates is then drawn, giving a 1:1 ratio of positive to negative training examples. This preserves more `OR` and `AND` examples than uniform class balancing while raising the relative share of the rare relations. The exponent 0.85 was chosen heuristically. It was not tuned on the development set, and no sensitivity analysis or sampler ablation was performed. The budget used here is given in Section 4.2.

3.3 Model variants and fine-tuning

Three pretrained GLiNER2 [8] span models are compared: `gliner2-base-v1`[1], `gliner2-large-v1`[2], and `gliner2-multi-v1`[3]. All models are adapted with Low-Rank Adaptation (LoRA), so only a small fraction of parameters is updated.

The Multilingual variant is included not as an evaluation of cross-lingual RE, since Chia is English-only, but as a comparison with the English Base checkpoint under the same English benchmark. This comparison is motivated by potential deployment in multilingual clinical settings, but the experiment does not isolate multilinguality itself as the cause of any observed performance difference.

For comparability, all three models are fine-tuned under the same training configuration, given in Section 4.3.

4 Experiment Setting

4.1 Dataset

Experiments use Chia, a manually annotated corpus of clinical trial eligibility criteria introduced by Kury et al. [4]. The original resource contains 12,409 eligibility criteria from 1,000 interventional Phase IV studies, with 41,487 distinctive entities from 15 entity types and 25,017 relationships from 12 relationship types. Chia aligns its clinical domains with the Observational Medical Outcomes Partnership (OMOP) Common Data Model and represents each criterion as an annotation graph that can be transformed into Boolean logic.

Eleven relation types are targeted. `Has_value` links a clinical concept to its value, `Has_qualifier` attaches a descriptive modifier, `Has_temporal` a temporal restriction, and `Has_index` connects a temporal expression to its reference point. `Has_negation`, `Has_multiplier`, and `Has_mood` connect the corresponding constructs, and `Has_context` links an entity to an observation used as context. `Subsumes` covers dependency, clarification, example, and specification between concepts, while `AND` and `OR` encode Boolean combinations. Chia also defines `Has_scope`, which the post-processed *Without Scope* variant used here excludes.

Chia is structurally difficult for pairwise RE. `OR` annotations can hold more than two arguments: 29.9% of the original `OR` relations connect more than two entities, and the largest reported group contains 25. Converting these into pairwise decisions multiplies the number of `OR` edges (Table 1). The corpus also permits nested, overlapping, and discontinuous annotations, so the relation arguments are more complex than in flat mention-based corpora.

At the criterion-conversion stage, criteria with no retained in-schema entity are excluded because they cannot yield reference-anchored relation candidate pairs. This leaves 10,767 of the 12,409 original criteria, partitioned into 7,714 training, 1,052 development, and 2,001 test criteria. Criteria originating from the same

[1] https://huggingface.co/fastino/gliner2-base-v1
[2] https://huggingface.co/fastino/gliner2-large-v1
[3] https://huggingface.co/fastino/gliner2.5-multi-v1

Table 1: Positive pairwise relation instances by partition. Train, Dev, and Test denote the training, development, and test partitions, respectively. The two logical relations are listed separately because pairwise expansion of their multi-argument groups inflates their counts.

Relation	Train	Dev	Test	Total
Has_value	2,671	384	750	3,805
Has_qualifier	2,549	385	598	3,532
Has_temporal	2,409	321	599	3,329
Subsumes	1,452	202	396	2,050
Has_index	649	79	171	899
Has_negation	616	78	149	843
Has_multiplier	439	61	132	632
Has_mood	347	74	102	523
Has_context	168	16	36	220
AND	2,646	340	688	3,674
OR	10,538	1,685	2,957	15,180
Total	**24,484**	**3,625**	**6,578**	**34,687**

ClinicalTrials.gov study record, identified by a unique ClinicalTrials.gov identifier (National Clinical Trial (NCT) number), are kept in the same partition.

4.2 Data preparation

The original brat annotations are converted into criterion-level JSONL records. Relation direction is preserved, and multi-argument logical annotations are expanded into pairwise relation instances. For example, a three-argument logical group OR(A,B,C) is represented by the pairwise relations OR(A,B), OR(A,C), and OR(B,C). This representation preserves the pairwise connections but no longer represents the original three-argument group as a single structure. Discontinuous mentions are represented by their minimum enclosing span because the evaluated span architecture operates on continuous intervals. In other words, when an annotated mention consists of several separated text segments, we use the shortest continuous text span that covers all of those segments.

After pairwise conversion, positive relation instances are distributed as shown in Table 1. The counts cover the full partitions before tempered sampling and exclude NA candidate pairs. OR alone accounts for 15,180 of the 34,687 positive instances, and the two logical relations together for 54%. This is the skew the sampler of Section 3.2 is built to temper.

Candidate pairs are then constructed from the reference annotations. The development partition yields 21,855 candidate pairs, 3,625 positive and 18,230 NA; the held-out test partition yields 42,091, of which 6,578 are positive and 35,513 NA. A few pairs carry more than one relation label, 14 in training, none in development and 5 in the test partition, and these expand to 28 training and 10 test instances. The tempered sampler of Section 3.2 is run with a budget of 6,000 positive instances and 6,000 NA candidates, giving 12,000 training pairs.

4.3 Models and training configuration

The Base, Large, and Multilingual checkpoints contain about 208.5M, 486.4M, and 307.1M parameters, and their LoRA adapters add about 1.64M, 3.97M, and 1.65M trainable parameters. LoRA targets the encoder query, key, and value projections together with the task-specific projection, classification, and count-prediction layers; the model-specific target-module lists are given with the experiment configuration in the project repository. The shared training configuration is summarised in Table 2. Experiments ran on a single NVIDIA GeForce RTX 4070 Ti SUPER with 16 GB of memory, using PyTorch 2.13.0 with CUDA 12.6, GLiNER2 1.3.2, Transformers 5.14.1, and PEFT 0.20.0.

Table 2: Hyperparameters used for parameter-efficient fine-tuning.

Hyperparameter	Value	Hyperparameter	Value
Training examples	12,000	Warmup steps	150
Fine-tuning method	LoRA	LR scheduler	Linear
Epochs	2	Max gradient norm	1.0
Batch size	4	LoRA rank (r)	8
Gradient accumulation	4	LoRA alpha	16
Effective batch size	16	LoRA dropout	0.0
Learning rate	5×10^{-4}	Training seed	20260810
Weight decay	0.01	Pair-construction seed	20260830

4.4 Evaluation metrics

Relation classification is evaluated with micro-averaged Precision, Recall, and F1, computed over the eleven relation classes. NA marks the absence of a relation rather than a relation type, and it is excluded because negative pairs outnumber positive ones by roughly five to one and would otherwise dominate the metric.

Because candidate spans and types are fixed by the reference annotations, a true positive requires the correct relation label for the correct ordered candidate pair. The confidence threshold is applied to the model-reported confidence of the predicted class: a non-NA prediction below the threshold is mapped to NA, while a predicted NA is retained. Thresholds are selected on the development set by sweeping predefined candidate values, with a finer search around the best-performing values, then fixed and applied unchanged to the held-out test partition. We also report macro-F1, the unweighted mean of the per-relation F1 scores. Unlike micro-F1 it gives equal weight to frequent and rare relations, which is the complementary view needed under this class distribution.

5 Results

Table 3 reports both partitions. Large is the strongest model on the held-out test partition at 0.7510 micro-F1, followed by Base at 0.7260 and Multilingual at 0.7039.

Between the two partitions Base changes little, from 0.7230 to 0.7260 micro-F1, while Large falls from 0.7628 to 0.7510 and Multilingual from 0.7113 to 0.7039. The ranking is preserved on unseen test data.

Table 4 breaks the Large model's test performance down by relation.

The strongest scores are for Has_index, Has_value, and Has_qualifier; Has_context and AND are the weakest. Support alone does not account for this. Has_index reaches 0.934 on 171 positives, while OR, the largest class at 2,957, reaches 0.734 and AND, at 688, reaches 0.538. The low Has_context score should be read cautiously, since only 36 positive examples occur in the test partition.

Inference over the 42,091 test candidate pairs took about 15.8 min for Base, 30.1 min for Large, and 15.7 min for Multilingual,

Table 3: Relation classification results on the development and held-out test partitions, with NA excluded from P/R/F1. Thresholds were selected on the development partition and applied unchanged to the test partition: 0.990 (Base), 0.995 (Large), 0.980 (Multilingual).

Model	Split	P	R	Micro-F1	Macro-F1
Base	dev	0.7043	0.7426	0.7230	0.6778
	test	0.7167	0.7355	0.7260	0.6952
Large	dev	**0.7393**	**0.7879**	**0.7628**	**0.7322**
	test	**0.7351**	**0.7676**	**0.7510**	**0.7494**
Multilingual	dev	0.6837	0.7412	0.7113	0.6461
	test	0.6876	0.7209	0.7039	0.6669

Table 4: Per-relation F1 scores and support of the Large model on the held-out test partition.

Relation	F1	*n*	Relation	F1	*n*
`Has_value`	0.873	750	`Has_negation`	0.827	149
`Has_qualifier`	0.859	598	`Has_multiplier`	0.818	132
`Has_temporal`	0.759	599	`Has_mood`	0.840	102
`Subsumes`	0.644	396	`Has_context`	0.417	36
`Has_index`	0.934	171	`AND`	0.538	688
			`OR`	0.734	2,957

or about 44.4, 23.3, and 44.6 candidate pairs per second on the hardware in Section 4.3.

6 Discussion

The three variants reach 0.7039–0.7510 micro-F1 on the held-out test partition, with the same ordering on both partitions. On the English-only Chia benchmark, Multilingual reaches 0.7039 compared with 0.7260 for Base. This observed difference does not establish multilinguality as its cause, and Chia cannot evaluate cross-lingual RE capability.

Three limitations constrain interpretation. Fixed mention spans and types isolate relation classification from upstream detection errors, so the scores are not end-to-end results from raw text. No external RE system is evaluated, so the results support comparison only among the tested variants under this protocol. Finally, the high selected thresholds (0.980–0.995) suggest that confidence values are concentrated near the top of their range, leaving calibration for future work.

Residual difficulty comes from several sources. Pairwise expansion alters the original structure of multi-argument `OR` groups, while the tempered sampler reduces imbalance without reconstructing them. `Subsumes` covers several semantic dependencies under one label, whereas `Has_context` is rare and therefore unstable.

The models run on a single consumer GPU and avoid dependence on third-party inference services, although local deployment still requires appropriate security and validation.

7 Conclusion

We evaluate biomedical relation extraction as reference-anchored relation classification over eleven semantic and logical relation types and compare three compact schema-driven variants under one parameter-efficient fine-tuning protocol.

Large achieves the highest held-out micro-F1 of 0.7510. Performance varies substantially across relation types, reflecting logical structure, semantic ambiguity, and data scarcity rather than frequency alone.

Future work will consider stronger candidate generation, explicit modelling of logical groups, external relation-classification baselines, alternative sampling strategies, relation-specific calibration, additional biomedical RE corpora, and locally deployed clinical information extraction pipelines.

Acknowledgements

This work was supported by the Slovenian Research Agency. Funded by the European Union. UK participants in Horizon Europe Project PREPARE are supported by UKRI grant number 10086219 (Trilateral Research). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA) or UKRI. Neither the European Union nor the granting authority nor UKRI can be held responsible for them. Grant Agreement 101080288 PREPARE HORIZON-HLTH-2022-TOOL-12-01.

References

[1] Elisa Bassignana and Barbara Plank. 2022. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 67–83.

[2] Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu. 2020. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, 88–99. doi:10.18653/v1/2020.coling-main.8.

[3] Jitendra Jonnagaddala and Zoie Shui-Yee Wong. 2025. Privacy preserving strategies for electronic health records in the era of large language models. *npj Digital Medicine*, 8, 1, 34. doi:10.1038/s41746-025-01429-0.

[4] Fabrício Kury, Alex Butler, Chi Yuan, Li-Heng Fu, Yingcheng Sun, Hao Liu, Ida Sim, Simona Carini, and Chunhua Weng. 2020. Chia, a large annotated corpus of clinical trial eligibility criteria. *Scientific Data*, 7, 1, 281. doi:10.1038/s41597-020-00620-0.

[5] Yanis Labrak, Mickael Rouvier, and Richard Dufour. 2024. A zero-shot and few-shot study of instruction-finetuned large language models applied to clinical and biomedical tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2049–2066.

[6] Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N. Arighi, and Zhiyong Lu. 2022. BioRED: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23, 5, bbac282. doi:10.1093/bib/bbac282.

[7] Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4904–4917. doi:10.18653/v1/2022.acl-long.337.

[8] Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. 2025. GLiNER2: an efficient multi-task information extraction system with schema-driven interface. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Suzhou, China, 130–140. https://aclanthology.org/2025.emnlp-demos.10/.

[9] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5364–5376. doi:10.18653/v1/2024.naacl-long.300.

The Role of Artificial Intelligence in Early Moisture Risk Detection for Wooden Buildings

Kristina Piiarska
University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies
Koper, Slovenia
kristina.piiarska@famnit.upr.si

Aleksandar Tošić
University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies
Koper, Slovenia
InnoRenew CoE
Izola, Slovenia
aleksandar.tosic@upr.si
aleksandar.tosic@innorenew.eu

Pau Brunet
Forest Science and Technology Centre of Catalonia
Solsona, Spain
pau.brunet@ctfc.cat

Abstract

The operation and maintenance of modern buildings increasingly relies on sensor networks that continuously collect large amounts of environmental and structural monitoring data. In wooden buildings, continuously monitoring these measurements remains challenging, allowing moisture-related problems to go unnoticed until costly intervention is required. This study investigates whether a fine-tuned Granite Time Series Foundation Model can support early moisture risk detection by forecasting future moisture development and providing automated alerts. A separate Granite model was fine-tuned for each building using moisture and environmental measurements collected from the InnoRenew CoE building in Slovenia and the Forestry HUB building in Spain. Performance was evaluated on unseen seasonal test data and compared with daily-naive, weekly-naive, Linear Regression, and Random Forest baselines. The Granite model mostly outperformed the naive methods and Random Forest, while achieving MAE and RMSE comparable to Linear Regression. Despite the similar numerical accuracy, Granite better captured the temporal patterns and fluctuations present in the ground-truth measurements. The results demonstrate the potential of pre-trained time-series models for automated moisture monitoring and early risk alerts in wooden buildings.

Keywords

wood moisture, time-series forecasting, alert systems, building monitoring

1 Introduction

Wood has become an increasingly important construction material due to its renewable nature and widespread use in sustainable construction. As the number of wooden buildings continues to increase, ensuring their long-term durability and reliable operation becomes increasingly important. Among the various factors influencing the service life of wooden structures, moisture is one of the most critical. Elevated moisture levels can lead to fungal decay, mould growth, and other degradation processes that reduce structural performance and may ultimately require costly maintenance or repair.

To mitigate these risks, modern wooden buildings are increasingly equipped with sensor networks that continuously monitor wood moisture content together with environmental variables such as temperature and relative humidity. These systems generate large volumes of valuable information. However, collecting data alone does not guarantee effective preventive maintenance. In practice, building owners and facility managers do not continuously inspect thousands of sensor measurements or identify subtle temporal patterns that may indicate the early stages of moisture-related problems. Consequently, important developments may remain unnoticed despite being recorded by the monitoring infrastructure.

Artificial intelligence offers an opportunity to transform building monitoring from passive data collection into an active decision-support process. Recent advances in time-series forecasting enable learning complex temporal relationships from historical observations and forecasting future system behaviour. Such systems are not intended to replace human expertise, but instead to continuously analyse incoming measurements and automatically draw attention to situations that may require further inspection.

If successful, this approach could improve preventive maintenance by allowing building managers to identify developing moisture-related risks before visible damage occurs, reducing maintenance costs while improving building durability. Beyond wooden buildings, similar methodologies could support automated monitoring and early warning in other structural health monitoring applications where large volumes of sensor data are continuously collected, including bridges, historical buildings, industrial facilities, and other critical infrastructure.

In this paper, we investigate whether a fine-tuned Granite Time Series Foundation Model can support early moisture risk detection in wooden buildings through short-term forecasting of wood moisture content. A separate Granite model is fine-tuned and evaluated for each building: the InnoRenew CoE building in Slovenia and the Forestry HUB building in Spain. The forecasting performance is compared with statistical and machine-learning baseline methods to assess the suitability of this approach for practical building monitoring applications.

The remainder of this paper is organized as follows. Section 2 reviews related work on wood moisture monitoring, predictive maintenance, and time-series forecasting. Section 3 describes the study buildings, datasets, data preparation, forecasting models, and experimental methodology. Section 4 presents the experimental results and compares the proposed approach with the baseline methods. Finally, Section 5 concludes the paper by summarizing the main findings, discussing the limitations of the study, and outlining directions for future research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).

2 Related Work

Previous studies have established the relationship between wood moisture and long-term durability, showing that prolonged exposure to elevated moisture levels increases the risk of fungal decay and other degradation processes [4]. Other works have focused on continuous monitoring of timber buildings using embedded sensors, wireless monitoring systems, and hygrothermal modelling to analyse moisture evolution under real operating conditions [8, 11]. More recently, machine learning has also been applied to predict moisture behaviour in timber structures, for example through Support Vector Regression models for long-term hygrothermal performance under future climate scenarios [2]. Together, these studies demonstrate the importance of continuous moisture monitoring for preserving the durability of wooden buildings. However, they primarily focus on moisture measurement, retrospective analysis, or long-term prediction rather than operational forecasting capable of automatically identifying developing moisture-related risks from continuously collected sensor data. In contrast, our work investigates whether a pre-trained time-series foundation model can forecast future moisture development and provide automated early warnings for building monitoring.

Artificial intelligence has become an increasingly important tool for predictive maintenance and forecasting in buildings. Bouabdallaoui et al. [3] proposed a machine-learning framework for predictive maintenance of building facilities, while Guo et al. [7] demonstrated that deep learning models can accurately forecast multivariate building energy consumption. Kramar and Alchakov [9] further showed that forecasting performance depends strongly on the ability of models to capture recurring seasonal patterns. Although these studies demonstrate the potential of artificial intelligence for analysing building-related time series, they either address different application domains or rely on task-specific forecasting models. Our work instead focuses on operational moisture monitoring in wooden buildings and investigates whether a pre-trained foundation model can be adapted through fine-tuning rather than developing a dedicated forecasting architecture from scratch.

Recent advances in time-series foundation models have introduced a new generation of forecasting models pre-trained on large and diverse collections of time-series data. Chronos reformulated forecasting as a language modelling problem [1], while Das et al. [5] proposed a decoder-only foundation model trained on heterogeneous datasets. Ekambaram et al. [6] introduced Tiny Time Mixers, demonstrating that lightweight pre-trained models can achieve competitive forecasting performance with substantially lower computational requirements. More recently, Park et al. [10] demonstrated that fine-tuning foundation models significantly improves forecasting performance for building energy systems compared with zero-shot inference. These studies confirm the strong potential of foundation models for time-series forecasting, yet their application to operational wood moisture forecasting remains largely unexplored. Building upon this recent progress, our work evaluates the IBM Granite Time Series Foundation Model for forecasting wood moisture content using long-term monitoring data collected from two wooden buildings and compares its performance with conventional statistical and machine-learning forecasting methods.

3 Methodology

3.1 Data Preparation

The experiments presented in this work were conducted using monitoring data collected from two wooden buildings: the InnoRenew CoE building in Slovenia and the Forestry HUB building in Spain. Both buildings are equipped with monitoring systems that continuously measure the condition of wood.

The preprocessing pipeline was designed to create a consistent forecasting dataset. After hourly aggregation, the measurements from each sensor were organised into independent multivariate time series. Wood moisture content was selected as the forecasting target, while temperature, relative humidity, absolute humidity, and temporal variables were used as model inputs. Since temporal variables are cyclical by nature, the month, hour of the day, and day of the year were encoded using sine and cosine transformations.

Each sensor time series was divided into four seasonal subsets corresponding to winter, spring, summer, and autumn. For every season, the final 96 hours were reserved as the test period, while the preceding 768 hours were used for validation and the remaining observations for training. To make better use of the limited seasonal data, the 512-hour input context required for testing was taken from the last 512 hours of the validation period immediately preceding the test horizon.

The available datasets also contained periods with missing observations. The InnoRenew CoE dataset was missing approximately two weeks of measurements during autumn, while the Forestry HUB dataset was missing almost one month of summer observations. Consequently, fewer samples were available for training and evaluation in these seasons, which affected the forecasting performance.

3.2 Granite Time Series Foundation Model

The primary forecasting model evaluated in this study is the IBM Granite Time Series Foundation Model (Granite TTM), a pre-trained deep learning model designed for multivariate time-series forecasting. Instead of being trained from scratch for a single application, Granite is first pre-trained on a large collection of time-series datasets and can then be adapted to a specific forecasting task through fine-tuning. This approach enables the model to benefit from previously learned temporal representations while requiring only a relatively small amount of task-specific training data. The experiments were performed using the publicly available IBM Granite Time Series Foundation Model (`ibm-granite/granite-timeseries-ttm-r2`, revision `512-96-ft-r2.1`), which supports a context length of 512 hours and a prediction horizon of 96 hours.

3.3 Baseline Models

To evaluate the forecasting performance of the Granite model, its predictions were compared with four baseline approaches representing different levels of forecasting complexity: two persistence-based methods and two commonly used machine-learning algorithms.

The first baseline is the *daily-naive* model, which assumes that future moisture values follow the same pattern observed 24 hours earlier. Each prediction is obtained by repeating the corresponding observation from the previous day, providing a simple reference for short-term forecasting.

The second baseline is the *weekly-naive* model, which predicts future values using observations recorded at the same time one week earlier. Compared with the daily-naive approach, it captures longer repeating patterns associated with weekly environmental cycles.

The third baseline is a multivariate Linear Regression model, which estimates future wood moisture as a linear combination of the input variables. All input variables were standardized using the mean and standard deviation computed from the training data, and the same transformation was applied to the validation and test data to ensure consistent feature scaling.

The final baseline is a Random Forest regressor consisting of 500 decision trees with a maximum depth of 20 and a minimum leaf size of two samples. Random Forest models nonlinear relationships by combining the predictions of multiple decision trees, making it capable of capturing more complex interactions between wood moisture, environmental conditions, and temporal variables than a linear model.

All baseline models were evaluated using the same training, validation, and test datasets and generated forecasts for the same 96-hour prediction horizon. This ensured that differences in forecasting performance were attributable solely to the forecasting methods.

3.4 Experimental Design

Three experiments were conducted. The first experiment evaluated the Granite model against the daily-naive, weekly-naive, Linear Regression, and Random Forest baselines using the InnoRenew CoE dataset. The second experiment applied the same experimental methodology to the Forestry HUB dataset, allowing the forecasting performance to be evaluated under a different building and climatic environment. The third experiment evaluated whether the proposed forecasting system can detect an unexpected increase in wood moisture using an artificial moisture increase introduced into the test data.

For the first and second experiments, the Granite model was fine-tuned for 100 epochs, while validation was performed using a sliding-window approach with multiple overlapping input-target windows. The checkpoint achieving the lowest validation loss was selected for the final evaluation. The selected checkpoint was then evaluated on the final 96-hour test horizon using the last 512-hour historical context from the validation period. This procedure was repeated independently for winter, spring, summer, and autumn to evaluate forecasting performance across different environmental conditions.

For the third experiment, an artificial moisture increase was introduced only into the measured values of the final 96-hour winter test period. Since neither monitored building had abnormally high moisture levels during the observation period, no real moisture event suitable for evaluating the proposed alerting procedure was available. The artificial sequence was therefore generated by adding a smooth nonlinear drift based on a Smoothstep interpolation function together with small temporally correlated fluctuations to the original measurements, simulating a gradually developing moisture increase. At each forecast step, the absolute difference between the Granite prediction and the modified observation was calculated. A deviation alert was raised when this difference exceeded the selected threshold for at least three consecutive hours, while a separate alert was generated when the modified moisture value exceeded the predefined safety threshold.

3.5 Evaluation Metrics

Forecasting performance was evaluated using two widely adopted regression metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Both metrics quantify the difference between the predicted and observed wood moisture content over the 96-hour forecasting horizon. For each forecasting model, MAE and RMSE were computed separately for every season. In addition, overall MAE and RMSE values were obtained by averaging the results across all seasonal forecasting experiments, providing an overall comparison of model performance.

4 Results

Tables 1 and 2 summarize the overall forecasting accuracy of all evaluated methods, while Figure 1 presents the corresponding seasonal forecasts for both datasets. On the InnoRenew CoE dataset, Granite consistently outperformed both persistence-based baselines and achieved forecasting accuracy comparable to the classical machine learning models. Although Linear Regression obtained the lowest overall MAE and RMSE, the difference compared with Granite was small. The seasonal forecasts further confirm these results, showing that Granite consistently reproduced both the overall trend and short-term moisture variations across all seasons.

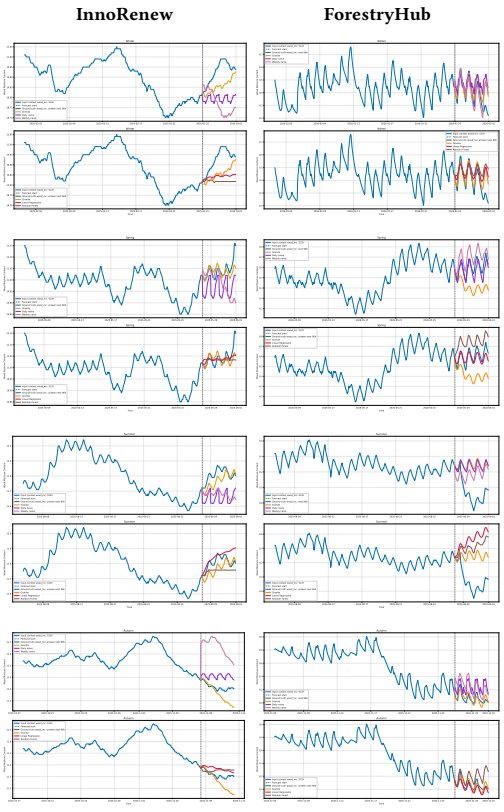


Figure 1: Seasonal forecasting results for the InnoRenew and ForestryHub datasets.

The ForestryHub dataset proved more challenging due to the higher variability of the measured wood moisture. Granite produced accurate forecasts during winter and autumn and remained competitive during spring. The main exception was summer, where the measured moisture dropped abruptly immediately after the beginning of the forecasting horizon (Figure 1), creating a

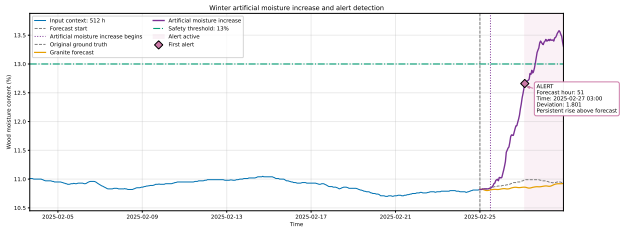


Figure 2: Artificial moisture increase experiment.

pattern that was not present in the preceding input context. This season also contained the smallest amount of available training data, which likely contributed to the higher prediction errors. Consequently, Granite ranked behind the persistence-based baselines in the overall MAE and RMSE reported in Table 2. Nevertheless, the seasonal forecasts indicate that Granite followed the overall evolution of the moisture signal more closely than the baseline methods, which often repeated previously observed patterns or gradually diverged from the measured trend.

Table 1: Overall forecasting performance on the InnoRenew CoE dataset.

Method	MAE	RMSE
Granite	0.0488	0.0643
Daily naive	0.1137	0.1243
Weekly naive	0.1740	0.2149
Linear Regression	0.0483	0.0627
Random Forest	0.0572	0.0745

Table 2: Overall forecasting performance on the ForestryHub dataset.

Method	MAE	RMSE
Granite	0.1551	0.2205
Daily naive	0.1538	0.2123
Weekly naive	0.1522	0.1879
Linear Regression	0.1784	0.3135
Random Forest	0.2055	0.2810

The artificial moisture increase experiment further demonstrated the practical applicability of the proposed forecasting framework for building monitoring. As illustrated in Figure 2, the modified measurements gradually deviated from the Granite forecast and eventually exceeded the predefined safety threshold. By continuously comparing predicted and measured moisture values, the monitoring procedure successfully detected the developing moisture increase and generated an alert before the end of the forecasting horizon. The figure highlights the first detected alert, which indicates the moment when the predefined alert conditions were first satisfied.

5 Conclusion

This work investigated the applicability of the Granite TinyTimeMixer foundation model for short-term wood moisture forecasting using two independent datasets collected from wooden buildings. The proposed approach was evaluated against naive forecasting methods and classical machine learning models using seasonal forecasting experiments with a 96-hour prediction horizon. The experimental results showed that Granite consistently ranked among the best-performing forecasting methods across both datasets. On the InnoRenew CoE dataset, its forecasting accuracy closely matched the best-performing classical machine learning model, while on the ForestryHub dataset it remained competitive despite the higher variability of the measurements. In addition to the numerical evaluation, the seasonal forecast curves showed that Granite generally reproduced the temporal evolution of wood moisture more faithfully than the baseline methods. The artificial moisture increase experiment showed that the proposed approach can detect abnormal moisture development and issue an alert when predefined conditions are met.

Overall, the results indicate that pretrained foundation models represent a promising alternative for wood moisture forecasting and that their predictions can be effectively combined with threshold-based monitoring to support the early identification of abnormal moisture behaviour. Future work should investigate training on larger datasets, including additional buildings and longer observation periods, to further improve the robustness and generalization of forecasting.

References

[1] Abdul Fatir Ansari et al. 2024. Chronos: learning the language of time series. *Transactions on Machine Learning Research*. https://arxiv.org/abs/2403.07815.

[2] Naman Bansal, Maurice Defo, and Michael Lacasse. 2021. Application of Support Vector Regression to the prediction of the long-term impacts of climate change on the moisture performance of wood frame and massive timber walls. *Buildings*, 11, 5, 188. doi:10.3390/buildings11050188.

[3] Yassine Bouabdallaoui, Zoubeir Lafhaj, Pascal Yim, Laure Ducoulombier, and Belkacem Bennadji. 2021. Predictive maintenance in building facilities: a machine learning-based approach. *Sensors*, 21, 4, 1044. doi:10.3390/s21041044.

[4] Christian Brischke and Gry Alfredsen. 2020. Wood-water relationships and their role for wood susceptibility to fungal decay. *Applied Microbiology and Biotechnology*, 104, 3781–3795. doi:10.1007/s00253-020-10479-1.

[5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR. https://arxiv.org/abs/2310.10688.

[6] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. 2024. Tiny Time Mixers (TTMs): fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Advances in Neural Information Processing Systems (NeurIPS)*. NeurIPS Foundation. https://arxiv.org/abs/2401.03955.

[7] Jing Guo, Penghui Lin, Limao Zhang, Yue Pan, and Zhonghua Xiao. 2023. Dynamic adaptive encoder-decoder deep learning networks for multivariate time series forecasting of building energy consumption. *Applied Energy*, 350, 121803. doi:10.1016/j.apenergy.2023.121803.

[8] Steven Kordziel, Samuel V. Glass, Shiling Pei, Samuel L. Zelinka, and Paulo Cesar Tabares-Velasco. 2018. Moisture monitoring and modeling of mass timber building systems. In *Proceedings of the World Conference on Timber Engineering (WCTE)*. World Conference on Timber Engineering. https://api.semanticscholar.org/CorpusID:198425188.

[9] Vadim Kramar and Vasiliy Alchakov. 2023. Time-series forecasting of seasonal data using machine learning methods. *Algorithms*, 16, 5, 248. doi:10.3390/a16050248.

[10] Young-Jin Park, François Germain, Jing Liu, Ye Wang, Toshiaki Koike-Akino, Gordon Wichern, Navid Azizan, Christopher Laughman, and Ankush Chakrabarty. 2025. Probabilistic forecasting for building energy systems using time-series foundation models. (2025). arXiv: 2506.00630 [cs.LG]. doi:10.48550/arXiv.2506.00630.

[11] Maryam Shirmohammadi, Marcus Strang, Paola Leardini, Kim Baber, and Adam Faircloth. 2025. Moisture monitoring of mass timber building – study of condition variation and building environment design. In *Proceedings of the World Conference on Timber Engineering (WCTE)*, 2589–2598. doi:10.52202/080513-0316.

E3 Atlas: An Autonomous LLM Agent for Curating and Serving a Departmental Knowledge Base

Drejc Pesjak*
drejc.pesjak@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Georgi Trajkov*
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Luis Rei
luis.rei@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

Institutional knowledge in research organizations is fragmented across shared folders, project archives, publication lists, and the memories of individual researchers. We present E3 Atlas, a deployed system in which a large language model (LLM) agent builds, verifies, and serves a knowledge base for the Department of Artificial Intelligence (E3) at the Jožef Stefan Institute. The agent runs on the open-source Hermes harness and combines several memory layers, the main one being a typed Markdown knowledge graph of people, projects, organizations, publications, and resources. Graph writes follow a verify-before-write methodology with per-node confidence and provenance, and every node is approved by a human. On a 485-question benchmark, Atlas far outperforms an ungrounded LLM (78.4% vs. 20.0% correct) but does not beat simpler grounded baselines: classical RAG reaches 80.6% and agentic file search 81.6%. On 50 questions whose answers the corpus does not support, Atlas gave a confident answer 26 times, while RAG declined all 50. We describe the architecture, the curation methodology, and the evaluation, and conclude that verify-before-write must be matched by verify-before-answer.

Keywords

LLM agents, knowledge graphs, knowledge management, retrieval-augmented generation, human-supervised curation

1 Introduction

Research departments accumulate knowledge faster than anyone documents it: staff and roles change, projects start and finish, and deliverables, papers, tools, models, and recorded talks are produced continuously. This knowledge is fragmented across shared folders, grant archives, mailing lists, and personal recollection. Newcomers onboard slowly, and simple questions ("who worked on emotion detection?", "which Horizon Europe projects are active?") require asking the right person rather than querying a system.

We present **E3 Atlas**, a deployed system that realizes this pattern for the Department of Artificial Intelligence (E3) of the Jožef Stefan Institute. A single always-on LLM agent maintains the department's knowledge base: a typed Markdown knowledge graph, prose summaries, and recurring intelligence briefings, all served to department members through an authenticated web portal and a Slack front end. Our contributions are:

(1) the architecture of a practical, multi-layer-memory agentic knowledge management system with built-in automation;
(2) a verify-before-write methodology with provenance tracking that keeps agent-written knowledge auditable.

2 Related Work

Retrieval-augmented generation (RAG) [5] addresses the *query-time* half of the information retrieval problem. GraphRAG [2] extends classical RAG by first building an entity graph over a corpus to better support global, query-focused summarization. RAG answers from whatever corpus it is given, while keeping that corpus complete, current, and consistent is usually left to humans. Recent work on LLM agents [10, 9] suggests a different division of labor: let an autonomous agent do the continuous curation, including collecting evidence, verifying facts, and writing structured knowledge, while humans supervise and correct. This is the approach E3 Atlas takes: corpus *construction and maintenance* is the agent's primary task, with retrieval as a downstream benefit.

We use a Knowledge Graph (KG) as an organizational substrate [3]. Our graph is deliberately lightweight, inspired by the LLM Wiki concept [4]: typed Markdown files with YAML frontmatter under version-controlled review, favoring human-readable, directly editable representations over formal semantics and specialized tooling.

The layered-memory design of E3 Atlas is influenced by agent memory architectures such as MemGPT [7] and mechanisms surveyed in [11]. The agent skills mechanism follows the emerging practice of agents maintaining procedural knowledge [1].

3 System Architecture

3.1 Agent Harness and Models

Figure 1 gives an overview of the system. E3 Atlas runs on *Hermes* [6], an open-source autonomous agent harness, deployed on a dedicated department virtual machine. The harness provides the tool loop (terminal, file manipulation, web search and extraction, sub-agent delegation), a multi-platform gateway (Slack, CLI, web

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.90

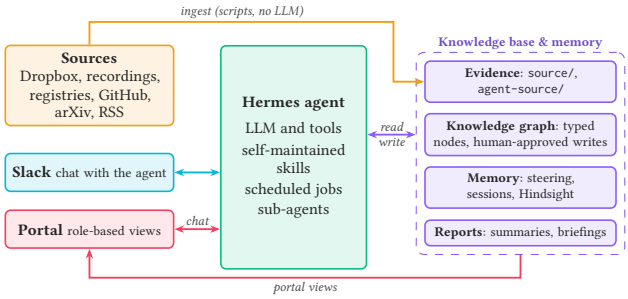


Figure 1: E3 Atlas architecture.

dashboard), scheduled jobs, and persistent state. The model used is Anthropic Claude, with an automatic fallback chain through OpenAI GPT and xAI Grok so that scheduled jobs do not stall on outages. On top of the stock harness we added the knowledge-base layout and its curation rules, the knowledge-graph schema, the memory layers, the briefing pipelines, the department portal, and all the related agent skills.

3.2 Knowledge Base Layout

The knowledge base separates evidence from derived knowledge:

- `source/` holds human-provided authoritative material (read-only to the agent);
- `source-text/` holds the text extracted from the provided documents (read-only to the agent);
- `agent-source/` holds raw evidence the agent collects (web pages, papers, external repositories);
- `knowledge-graph/` holds the durable entity nodes;
- `summaries/` and `briefings/` hold one-off reports and recurring intelligence products generated by the agent.

This separation helps make every derived claim traceable to evidence the agent did not author - a precondition for auditing machine-written knowledge.

3.3 The Knowledge Graph

The knowledge graph (KG) is a typed, interlinked corpus of plain Markdown files with the filesystem as the store, kept under Git version control: one file per node, folder-partitioned by node type. The following node types exist: person, project, publication, organization, meeting, infrastructure, programme, and resource with subtypes that include code repositories, datasets, models, and tools. Nodes are authored by an LLM agent that reads from the knowledge base source data folders (e.g. `source/`), or from sanctioned external registries, and writes claims with their associated provenance. Each file starts with YAML frontmatter: a machine-readable key-value block that includes:

- a globally unique slug, serving as the identifier for the node;
- relationship fields, with keys named for the relation itself (e.g. authors, funder, partners) and slugs of other nodes as the values;
- external identifiers (e.g. DOI, ORCID, CORDIS, URLs);
- sources (where claims came from);
- subject-descriptor field containing lists of topics or keywords.

The schema restricts which relationships each node type may carry. Most must resolve to an existing node, but a few may name people or organizations that deliberately get no node, such as external collaborators. Sources are paths to read-only evidence, registry URLs and identifiers, or plain-text references to confirmation by the administrator.

Subject-descriptor fields (topics, expertise, or keywords, depending on type) draw on a controlled vocabulary of research domains and methods declared in the schema, and the validator flags unregistered terms. Descriptors that the transcript pipeline emits for meeting nodes are free text.

After the frontmatter comes a short, type-specific prose body (typically 100–400 words) for human readers: the author abstract for publications, the funder's objectives for projects, a description plus hard facts (repository, licence, activity) for code, datasets, and models, and for people a summary of role and research focus with their national research-registry record, timestamped at retrieval.

Edges take two forms. *Hard links* are typed, directed slugs in the frontmatter, each written once on the side whose list is bounded and authoritative (a publication lists its authors and parent project, and a resource lists its maintainers). Inverse directions come from computed backlinks. *Soft links* are untyped `[[wikilinks]]` in the prose body. A static-site generator publishes the KG as a web viewer with backlinks and a graph view, and a hybrid lexical/vector index makes it searchable together with the underlying knowledge base.

3.4 Memory Layers

Four complementary memory layers keep the agent effective across sessions:

(1) *steering memory* provides a compact size-limited dynamic set of instructions, conventions, preferences, and behavioral rules that is injected into every turn;
(2) *session store*, a searchable archive of past conversations;
(3) *Hindsight* [8], a cross-session entity-centric memory backed by a relational store;
(4) *agent skills*, a set of self-authored and self-maintained procedural documents (e.g., how to compile papers in the IS template, how to query the national research registry, how to curate the knowledge graph).

Durable, curated departmental knowledge belongs in the knowledge graph, accumulated entity-centric knowledge in Hindsight, procedures in skills, and only compact behavioral rules and preferences in the capped steering memory.

3.5 Automation and Human Oversight

Automation differs by task. *Ingestion* of new data and maintenance run automatically as scheduled scripts that invoke no model. *Graph writes* are generated by the model but approved by a human, and *briefings* are fully autonomous.

Ingestion mirrors external sources such as Dropbox, converts PDF, slide, and Word documents to Markdown, and refreshes the retrieval index. It also searches public repositories (e.g., Zenodo, Hugging Face) and bibliographic indexes, and checks that each resource was authored by a department member under department funding. This makes new material *searchable*. The model-driven step that reads imported material and writes nodes is a separate job that a human starts, and every node it writes is approved by a human.

Maintenance jobs are script-only and follow a watchdog convention: silent on success, reporting only on change or failure. The simple example is the nightly automated backup of the knowledge graph as a compressed archive.

Briefings are the fully autonomous category. Six topic-scoped jobs run overnight to produce daily AI research and ecosystem briefings from sources including arXiv, X, RSS feeds, and blog posts. Their top-ranked items are aggregated into a single daily briefing delivered to the department Slack channel.

3.6 Web Interface

An authenticated web portal with role-based access gives department members an entry point to the knowledge graph, meeting transcripts and recordings, briefings, and generated summaries, plus a chat interface to the agent and administrative views of the agent and its memory. Views read the agent-maintained knowledge base directly, so new content appears without a separate publication step.

4 Curation Methodology

Letting an LLM write an institutional record raises an obvious risk: confident fabrication. The E3 Atlas methodology treats this as the central design constraint.

4.1 Verify before write and Provenance

A node may be minted only after its key facts are confirmed by authoritative evidence (publication, institutional page, project database, national research registry). A single authoritative source is the minimum requirement, but claims are cross-checked against multiple independent sources when possible. When evidence is partial, the agent must mint a stub with *'confidence: low'* and a *'needs-review'* tag rather than guess.

Every node lists its sources, derived claims in prose link back to evidence files or URLs. When sources disagree, a precedence rule decides. For example, a signed grant agreement overrides the proposal and its evaluation report on titles, consortium composition, and deliverables.

4.2 Schema governance

The schema itself evolves under explicit, numbered rules negotiated with human curators — e.g., a rule fixing how partner organizations are modeled (the participating unit becomes the node, and external contacts are kept inline rather than as person nodes, partly for data minimization). Rules are recorded in the schema document and enforced by audit scripts that check frontmatter validity, slug uniqueness, and dangling edges.

4.3 Corrections

Department members who identify gaps or mistakes can correct the KG through the chat interface, a process of **Human-in-the-loop correction** which requires the agent to find/obtain and verify supporting evidence. When these corrections reveal a procedural gap, the relevant agent skill is potentially semi-automatically improved so the same mistake is not repeated.

Automatic correction covers errors that scripts can detect and often fix deterministically, such as Markdown syntax errors, missing fields like ORCID, and missing or invalid provenance. These scripts are invoked by agent skills or listed in `AGENTS.md`, and keep the graph valid and free of dangling links.

5 Evaluation

5.1 Evidence Corpus and Knowledge Graph

Table 1 lists the largest document categories in the evidence corpus and the largest node types in the knowledge graph at the evaluation snapshot. The agent can additionally access 284 past conversations and 20,768 memory units in Hindsight.

Table 1: Top evidence-corpus categories by file count (left) and top knowledge-graph node types (right).

Corpus category	Files
Meeting/seminar transcript	292
Presentation	90
Proposal	89
Grant/consortium agreement	78
Evaluation	73
Deliverable	58

Node type	Nodes
Meeting	297
Publication	169
Project	153
Repository	114
Organization	77
Person	58

5.2 QA Benchmark

We sampled evidence passages across source types and document sizes, and an LLM (GPT-5.6-terra) wrote a question, gold answer, and evidence reference for each. Each type targets one capability. **Single-fact** (F) questions test retrieving one fact from one passage. **List/aggregation** (L) questions test collecting several items, usually within one document. **Relational** (R) questions test joining facts from two documents that share an entity. **Negative** (N) questions ask about plausible relations the corpus does not support, testing abstention (absence was checked against the retrieval index). A separate LLM verifier and a programmatic answer-leak check removed questions answerable without the corpus, ambiguous, or revealing their answer. Of 587 candidates, 487 were kept and 485 were judged for every system.

An author audit of 20 question–answer pairs found one incorrect gold answer (5%, 95% Wilson interval 0.9–23.6%), which was not corrected before scoring. Many questions reuse the vocabulary and identifiers of their source passage, so they do not reflect how people phrase questions.

Scoring. An LLM judge (GPT-5.6-sol) saw the question, gold answer, and system answer, but not the evidence, and scored 2 if all key facts were present and none contradicted, 1 if partially correct (e.g., an incomplete list), and 0 otherwise, ignoring phrasing and extra detail. Only 2 counts as correct. On negative questions a refusal scores 2 and any confident assertion 0. Such an assertion is unsupported by the corpus but not necessarily false. An independent model agreed with the judge on 20 of 20 sampled answers, but this check was not blind, and the judge shares a model family with the answering model.

5.3 Setup and Results

All systems answered with GPT-5.6-terra (temperature 1, fixed seed), without web access, once per question. The ungrounded LLM answered closed-book. Classical RAG used structure-aware, deduplicated 500-token chunks, hybrid BM25 and BGE-M3 retrieval with reciprocal rank fusion, and the top eight chunks. Agentic file search ran a fresh read-only Codex sandbox per question with ripgrep and sed. E3 Atlas ran in the Hermes harness (v0.20.5) with terminal, file, and memory tools and a 420 s limit. For Atlas, the graph was pinned to a fixed commit and the corpus to a fixed hash, and their fingerprint and that of the memory store were identical before and after the run. Memory writes and session search were disabled so nothing carried over between questions, and a tool gate blocked all 12 attempts to read benchmark or gold files.

Table 2 reports accuracy by question type. We compared systems with the two-sided exact McNemar test, which suits paired binary outcomes and uses only the questions on which two systems disagree, and applied a Holm correction to the three comparisons involving Atlas. On the 435 answerable questions, Atlas clearly outperformed the ungrounded baseline (adjusted $p < .001$). It did not differ significantly from RAG ($p = .121$) but was below agentic file search ($p = .019$, adjusted $p = .038$). Ten false refusals by Atlas account for most of the questions that only file search answered. Over all 485 questions Atlas ranks last of the three grounded systems, though neither difference is significant (RAG $p = .36$, file search $p = .052$). Whole-file access helped most on relational questions, where both agentic systems lead RAG by a wide margin.

Table 2: QA accuracy by question type (correct = judge score 2).

System (n)	F 237	L 98	R 100	N 50	Answ. 435	All 485
Ungrounded LLM	.169	.031	.160	.760	.136	.200
Classical RAG	**.924**	.643	.590	**1.000**	.784	.806
Agentic file search	.903	**.735**	**.880**	.440	**.860**	**.816**
E3 Atlas	.861	.704	.840	.460	.821	.784

RAG declined all 50 negative questions. Atlas declined 23, and 26 of its 27 other answers were confident, unhedged assertions, a pattern file search shares (26 of 28). The additional knowledge layers were rarely used: of the 426 answerable questions Atlas did not refuse, only 33 cited a knowledge-graph file, with similar accuracy with and without it (0.848 vs. 0.840), and the memory tool was never called although it was available. Reliable deployment therefore requires an answering procedure that specifies how to use the graph and memory, verifies provenance, and checks for absent evidence.

6 Discussion, Limitations, and Future Work

Why the approach is practical. E3 Atlas builds on information that the department already produces: project records in CORDIS and SICRIS, researcher profiles, publications, models, recorded meetings, and internally shared proposals. Much of this information is structured and accessible through databases or APIs, while the remaining material is generated by existing workflows. Consequently, deployment requires little change in staff behaviour: the agent performs the otherwise manual work of collecting, linking, and updating knowledge.

Trustworthiness and answering. Keeping evidence apart from derived knowledge, recording provenance and confidence, flagging uncertain claims as *'needs-review'*, and keeping corrections as procedures make curation auditable. These safeguards govern how knowledge is written, not how questions are answered: Atlas still answers confidently when the corpus lacks the information, so a verify-before-answer procedure is required.

Limitations. Deployment in a single department leaves transferability untested. The benchmark is synthetic, its gold answers are LLM-written and only partly audited, and each system ran once. Negative questions were verified as absent only against the RAG corpus, not against Atlas's larger substrate (graph, memory, and mirrored repositories), so some Atlas answers counted as fabrications may be supported there. The evaluation measures question answering, not the graph itself: schema validation guarantees well-formed nodes, not correct ones, and a stratified audit of 100 graph facts against independent sources is designed but not yet complete. Access control is enforced by user role at the portal. Propagating source-document permissions into the graph, memory, and answers is future work. Storing information about identifiable individuals also creates data-protection obligations.

Future work. We will add a verify-before-answer step that uses provenance and explicit evidence checks, complete the graph fact audit, run knowledge-graph and memory ablations, and evaluate on real user questions with independently reviewed gold answers. We also plan to package the system for other departments and to study which GDPR tasks (retention, access, correction, deletion) can be automated under human oversight.

E3 Atlas shows that a human-supervised LLM agent can build and serve a departmental knowledge base under an auditable curation methodology. It clearly outperforms an ungrounded model but not simpler grounded baselines, and its curated layers were rarely used when answering.

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency through the Young Researcher Grant (PR-13856), the research program Knowledge Technologies (P2-0103) and the Gravity project AI for Science (GC-0001) and European HE projects DataPact (GA 101189771) and Raido (GA 101135800). The authors acknowledge the use of Claude to support paper organization and language and grammar refinement. The authors reviewed and revised all AI-generated suggestions and take full responsibility for the content.

References

[1] Anthropic. 2025. Equipping agents for the real world with agent skills. Retrieved July 12, 2026 from https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills.

[2] Darren Edge et al. 2025. From local to global: a graph RAG approach to query-focused summarization. (2025). arXiv: 2404.16130 [cs.CL]. doi:10.48550/arXiv.2404.16130.

[3] Aidan Hogan et al. 2021. Knowledge graphs. *ACM Computing Surveys*, 54, 4, 71:1–71:37. doi:10.1145/3447772.

[4] Andrej Karpathy. 2026. LLM Wiki: a pattern for building personal knowledge bases using llms. GitHub Gist. Accessed 30 August 2026. (Apr. 2026). https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f.

[5] Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, 9459–9474.

[6] Nous Research. 2026. Hermes agent: an open-source autonomous agent harness. Retrieved July 12, 2026 from https://github.com/NousResearch/hermes-agent.

[7] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: towards LLMs as operating systems. (2023). arXiv: 2310.08560 [cs.AI]. doi:10.48550/arXiv.2310.08560.

[8] Vectorize. 2026. Hindsight: long-term memory for AI agents. Retrieved July 12, 2026 from https://github.com/vectorize-io/hindsight.

[9] Lei Wang et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18, 6, 186345. doi:10.1007/s11704-024-40231-1.

[10] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[11] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*, 43, 6, Article 155, (Sept. 2025), 47 pages. doi:10.1145/3748302.

Evaluating Large Language Model-Generated Explanations for Defective-Part Prediction in Injection Moulding Using Feature-Attribution Evidence

Seyed Iman Hosseini
iman.hosseini@ijs.si
Department for Artificial Intelligence, Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Lorenzo Seghesio
lorenzo.seghesio@kuleuven.be
Department of Mechanical Engineering, KU Leuven
Flanders Make@KU Leuven
Leuven, Belgium

Konstantinos Gryllias
konstantinos.gryllias@kuleuven.be
Department of Mechanical Engineering, KU Leuven
Flanders Make@KU Leuven
Leuven, Belgium

Dunja Mladenić
dunja.mladenic@ijs.si
Department for Artificial Intelligence, Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

This exploratory study examines LLM-generated explanations for injection-moulding defective-part predictions using local SHAP evidence. CatBoost detected two of ten held-out failures, both short-shot defects. GPT-5.4 mini, Gemini 3.1 Flash-Lite, and Mistral Devstral 2512 each explained both cases, yielding six explanations. One blinded LLM judge and one manufacturing expert assessed evidence faithfulness, attribution caution, and manufacturing-process relevance. The judge assigned both GPT explanations maximum attribution-caution scores, whereas the expert assigned minimum scores for unsupported process-value claims. This observed disagreement illustrates a potential limitation of automated explanation evaluation. With two cases of one defect type and one evaluator of each kind, the findings cannot establish a general ranking of LLMs.

Keywords

Explainable Artificial Intelligence, Large Language Models, injection moulding, model attribution, LLM as Judge

1 Introduction

Injection moulding is widely used for the large-scale manufacturing of plastic components ranging from standard to high-precision parts, but variations in process conditions can lead to quality defects and inconsistent production outcomes [6, 7]. In this study, the term *failure* refers to a production cycle that produces a defective or nonconforming part, rather than to a mechanical failure or breakdown of the injection-moulding machine. Traditional approaches such as Design of Experiments and Analysis of Variance help identify relevant process factors, but they may be limited in representing complex and nonlinear interactions among process variables [7]. Machine-learning studies now use XAI tools like SHAP and LIME to highlight key features and explain injection-moulding quality predictions [6, 7]. XAI-based predictive-quality studies have shown that process measurements, time-dependent information, and operator-adjustable parameters can provide useful evidence for explaining model outputs [6]. Local model-attribution methods are particularly useful because they indicate which variables most strongly influenced a prediction for an individual production cycle [1, 7]. However, model-attribution evidence explains the behavior of the predictive model rather than establishing the physical cause of a defect [1]. Furthermore, the direction of a feature attribution does not by itself indicate that the corresponding process measurement was high, low, abnormal, or outside an acceptable operating range [1]. Large Language Models (LLMs) provide a potential interface for translating technical model-attribution evidence into concise explanations for operators and quality-control personnel [1, 4, 8]. Previous studies have applied LLMs to fault diagnosis, process monitoring, and human–machine collaborative troubleshooting in industrial environments [4, 8]. Research on LLM-based explainability also suggests that LLMs can make technical evidence more accessible to non-specialist users [1]. Nevertheless, fluent natural-language explanations may introduce information that is absent from the supplied evidence, omit influential variables, reverse the direction of a contribution, or present model attribution as proof of physical causation [1, 4]. Therefore, readability alone is insufficient; generated explanations must also be evaluated for their faithfulness to the supplied evidence and for the caution with which they communicate it [1, 2]. Although LLMs are increasingly used to communicate model outputs and support fault diagnosis, limited attention has been given to whether their explanations accurately preserve local attribution evidence in manufacturing settings [1, 4, 8]. In particular, it remains unclear whether an LLM can correctly distinguish variables that push a prediction toward *Fail* from those that push it toward *Pass*, while avoiding unsupported statements about process values or physical root causes [1].

This exploratory study examines attribution errors and judge–expert disagreement in explanations of two short-shot cases detected among ten held-out failures. GPT-5.4 mini, Gemini 3.1 Flash-Lite, and Mistral Devstral 2512 produce six explanations, assessed by one blinded LLM judge and one manufacturing expert

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.95

for evidence faithfulness, attribution caution, and manufacturing-process relevance [2]. The findings describe these cases and do not establish a general model ranking.

2 Methodology and Experiments

This study evaluates the quality of LLM-generated explanations for injection-moulding defective-part prediction. The explanations are assessed by both an LLM judge and an independent manufacturing expert using three criteria: evidence faithfulness, attribution caution, and manufacturing-process relevance, as summarized in Table 3. The methodology consists of five stages: dataset loading and processing, predictive-model training and optimization, extraction of local feature-attribution evidence, generation of explanations by multiple LLMs, and evaluation using both the LLM judge and the human expert. The complete workflow is illustrated in Figure 1. The purpose of the study is not to identify the true physical cause of each defect, but to assess how accurately the generated explanations reflect the supplied attribution evidence, avoid unsupported claims, and remain relevant to manufacturing users.

2.1 Dataset

The study uses an injection-moulding dataset from the Korea AI Manufacturing Platform (KAMP) [3, 5], containing 5,232 production cycles and 35 variables describing machine, product, timing, pressure, and temperature conditions, together with the binary target *PassOrFail*. The dataset is highly imbalanced, with 5,172 passing cycles and 60 failures, corresponding to a failure rate of approximately 1.15%. The failures comprise 35 gas defects, 15 incomplete-moulding or short-shot defects, and 10 startup defects. Startup defects are removed because they occur during process stabilization and are treated as an expected operating condition. The resulting modelling dataset contains 5,222 cycles, including 50 failures. A reason-balanced train/test split is created using a fixed random seed. The test set contains five gas-defect cases, five short-shot cases, and approximately 20% of the passing cycles. This results in 4,178 training records with 40 failures and 1,044 test records with 10 failures. Production cycles are treated as independent samples, so time is not used in the split. Model tuning and threshold selection are performed using stratified four-fold cross-validation on the training set. Leakage-prone and identifier variables, including Reason, TimeStamp, row identifiers, production dates, and serial numbers, are excluded, leaving 29 predictor variables.

2.2 Predictive Model as Supporting Component

The predictive model is used to assign failure probabilities to production cycles and to provide the local model-attribution evidence used for explanation generation. Three gradient-boosting classifiers—CatBoost, LightGBM, and XGBoost—are compared. Because failures represent only a small proportion of the dataset, PR-AUC is used as the main model-selection metric, while failure recall and failure F1-score are reported as additional diagnostic measures. The hyperparameters of each classifier are optimized using Optuna and stratified four-fold cross-validation on the training set. Each model produces a probability that a production cycle will fail. A decision threshold is therefore needed to convert this probability into a final *Pass* or *Fail* prediction. Because failures are rare, the default threshold of 0.5 may miss many

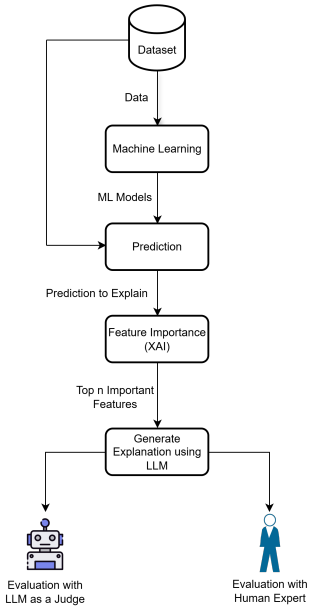


Figure 1: Overview of the study pipeline, from data preparation to LLM-based explanation and evaluation.

actual failure cases. For each model, the threshold is selected using validation predictions from the training data by maximizing the F2-score, which gives greater importance to detecting real failures while still considering false alarms. PR-AUC is used as the main model-selection metric because failures are rare in the dataset, and it measures how well the model identifies failure cases while accounting for false alarms. CatBoost achieved the highest mean cross-validation PR-AUC of 0.1565, followed by XGBoost with 0.1437 and LightGBM with 0.1083 as shown in Table 1. CatBoost was therefore selected as the downstream model according to the predefined cross-validation PR-AUC criterion. Of the ten observed failures in the test set, CatBoost correctly classified two as failures and missed the remaining eight. Only the two true-positive cases were selected for explanation generation because the study evaluates explanations of defective-part predictions made by the model. Both selected cases correspond to incomplete-moulding defects.

2.3 LLM-Based Explanation Generation

For each of the two correctly predicted failure cases in the held-out test set, the selected CatBoost model is used to calculate local SHAP feature attributions.The evidence package provided to the explanation-generating LLMs contains the most influential features, their signed attribution values, the direction in which each feature influenced the prediction, and a short manufacturing-readable description of each process variable. Features are ranked according to the absolute magnitude of their attribution, while the signed attribution is retained to indicate whether a feature pushed the prediction toward *Fail* or toward *Pass*. An example of the evidence package supplied to the explanation-generating LLM is presented in Table 2. The table shows the technical feature name, its manufacturing-readable description, its signed local attribution, and the direction in which it influenced the model prediction. A feature-description dictionary is used to translate technical variable names into terms that are easier to understand in an injection-moulding context. The attribution evidence is

Table 1: Performance comparison of the evaluated predictive models. Model selection was based on stratified cross-validation PR-AUC.

Model	CV PR-AUC	Test PR-AUC	Failure Recall	Failure F1-score	Selected
CatBoost	**0.1565**	**0.1711**	0.2000	0.1143	Yes
XGBoost	0.1437	0.0973	**0.4000**	**0.1509**	No
LightGBM	0.1083	0.0536	**0.4000**	0.0762	No

Note: Defective parts account for approximately 1.15% of the original dataset and 0.96% of the held-out test set.

treated as an explanation of the model's prediction rather than as evidence of the physical cause of the defect. Three LLMs—GPT-5.4 mini, Gemini 3.1 Flash-Lite, and Mistral Devstral 2512—are prompted to generate a short operator-facing explanation for each production cycle. Each model receives the same attribution evidence and feature descriptions to ensure a fair comparison. The prompt instructs the models to identify the most influential process variables, preserve the direction of their contributions, explain their relevance in manufacturing terms, and avoid unsupported statements about process-value levels, abnormal operating conditions, or definitive physical causes. Each of the two true-positive cases is explained by all three LLMs, resulting in six explanations for subsequent LLM-as-Judge and human-expert evaluation.

2.4 LLM-as-Judge Evaluation

The generated explanations are evaluated using a blinded LLM-as-Judge protocol. For each of the two true-positive failure cases, the judge receives the local model-attribution evidence and the three corresponding explanations. Model identities are hidden, and explanation order is randomized. Each explanation is scored from 1 to 5 for evidence faithfulness, attribution caution, and manufacturing-process relevance. The explanations are evaluated using a blinded LLM-as-Judge protocol with ~openai/gpt-mini-latest as the judge model. The overall score is calculated as the mean of these three scores. As shown in Table 4, the LLM judge assigned the GPT-5.4 mini explanations overall scores of 4.67 and 5.00, compared with 4.33 and 4.33 for Gemini 3.1 Flash-Lite and 4.33 and 3.00 for Mistral Devstral 2512. These scores describe the six evaluated explanations and should not be interpreted as estimates of the models' relative performance beyond these two cases.

2.5 Human-in-the-Loop Assessment

A central aim of this study is to compare automated LLM-as-Judge evaluation with human-in-the-loop assessment and examine their similarities and differences in judging LLM-generated manufacturing explanations. The human evaluator evaluates the six explanations generated for the two true-positive failure cases. For each case, the expert receives the same local feature-attribution evidence, manufacturing-readable feature descriptions, and corresponding LLM-generated explanations used in the automated evaluation. The expert scores each explanation from 1 to 5 according to the same three criteria used by the LLM judge: evidence faithfulness, attribution caution, and manufacturing-process relevance. While the LLM-as-Judge provides scalable and structured scores, the human-in-the-loop protocol introduces domain-informed judgment about process plausibility and practical relevance. Comparing the two evaluations allows us to investigate whether automated judging can approximate human assessment.

2.6 Evaluation Objective

The final objective is not to prove the true physical root cause of each failure, but to assess whether LLM-generated explanations are faithful to the local feature-attribution evidence and useful for human interpretation. This makes the study an evaluation-oriented contribution: it examines how generated manufacturing explanations can be assessed for grounding, caution, relevance, and actionability before being trusted in operator-facing settings.

2.7 Human Evaluation of LLM-Generated Explanations

The injection-moulding expert evaluated the explanations generated by the three LLMs, with results reported in Table 5. The expert assigned all six explanations the maximum evidence-faithfulness score, but differed substantially in attribution caution. GPT-5.4 mini and Mistral Devstral 2512 introduced unsupported claims about feature magnitudes, such as describing values as high or low, whereas Gemini 3.1 Flash-Lite avoided this error. Manufacturing relevance was reduced across all models because they stated that the available information could not provide a physical explanation of the failure. Although SHAP attributions alone are insufficient to determine the failure mechanism, the input features are measured process parameters and therefore contain physical information about the manufacturing process. The models also used the term "unshaped" instead of the standard injection-moulding term "short shot". GPT-5.4 mini was further penalized for referring to process parameters measurements as signals, while Gemini received a lower relevance score for part 116 due to imprecise process terminology. For these two cases, the expert assigned the Gemini explanations higher overall scores than the other explanations, principally because they avoided unsupported process-value claims. This case-specific assessment does not establish that Gemini generally produces better manufacturing explanations.

3 Results, Discussion, and Limitations

Across the two short-shot cases, the LLM judge assigned the highest overall scores to GPT-5.4 mini, whereas the manufacturing expert preferred Gemini 3.1 Flash-Lite. The clearest disagreement concerned attribution caution: both GPT explanations received 5 from the judge but 1 from the expert. The expert identified unsupported claims about higher injection speed, longer injection time, and higher injection pressure. Signed SHAP values describe the direction and magnitude of contributions to model output; they do not establish whether process measurements are high, low, abnormal, or physically responsible for a defect. The expert also identified unsupported magnitude claims in Mistral explanations, which the judge penalized inconsistently.

These observations illustrate how automated evaluation can overlook unsupported claims despite assigning high caution

Table 2: Example of the local model-attribution evidence supplied to the explanation-generating LLM for true-positive cases

Feature	Manufacturing-readable description	Attribution	Direction
`Plasticizing_Time`	Time required to melt and meter the material for the next shot through screw rotation	-0.7993	Toward *Pass*
`Max_Screw_RPM`	Maximum rotational speed of the screw during material preparation.	0.5960	Toward *Fail*

Interpretation: A positive attribution indicates that the feature pushed the model output toward *Fail*, whereas a negative attribution pushed it toward *Pass*. The attribution sign does not indicate whether the measured process value was high, low, or abnormal, and it does not establish physical causality.

Table 3: Criteria used for evaluating the LLM-generated explanations.

Criterion	Evaluation focus
Evidence faithfulness (EF)	Whether the explanation accurately reflects the provided model-attribution evidence.
Attribution caution (AC)	Whether the explanation avoids unsupported claims about process-value direction, abnormality, or physical causation.
Manufacturing-process relevance (MPR)	Whether the explanation is meaningful and relevant in the context of injection-moulding operations.

Table 4: LLM-as-Judge scores for the two true-positive cases.

Model	EF	AC	MPR	Overall
Case 1: Example 115				
GPT-5.4 mini	4	**5**	**5**	**4.67**
Gemini 3.1 Flash-Lite	4	**5**	4	4.33
Mistral Devstral 2512	4	4	**5**	4.33
Case 2: Example 116				
GPT-5.4 mini	**5**	**5**	**5**	**5.00**
Gemini 3.1 Flash-Lite	4	**5**	4	4.33
Mistral Devstral 2512	3	2	4	3.00

Note: EF = Evidence Faithfulness; AC = Attribution Caution; MPR = Manufacturing-Process Relevance. Bold values indicate the highest score within each true-positive example; ties are shown in bold.

Table 5: Manufacturing-expert evaluation of the LLM-generated explanations for the two true-positive cases.

Model	EF	AC	MPR	Overall
Case 1: Part 115				
GPT-5.4 mini	**5**	1	3	3.00
Gemini 3.1 Flash-Lite	**5**	**5**	**4**	**4.67**
Mistral Devstral 2512	**5**	1	**4**	3.33
Case 2: Part 116				
GPT-5.4 mini	**5**	1	3	3.00
Gemini 3.1 Flash-Lite	**5**	**5**	3	**4.33**
Mistral Devstral 2512	**5**	1	**4**	3.33

Note: EF = Evidence Faithfulness; AC = Attribution Caution; MPR = Manufacturing-Process Relevance. Scores range from 1 to 5. Bold values indicate the highest score for each criterion within a case.

scores. They do not establish a general ranking of explanation-generating models. The six explanations share only two underlying production cases of one defect type, assessed by one human expert and one LLM judge. Consequently, score differences are descriptive and cannot establish reliable model superiority or the prevalence of evaluator disagreement. Both cases were correctly detected failures, leaving explanation quality for incorrect predictions unexplored. Broader conclusions require more diverse cases, including false positives and false negatives, together with multiple experts and LLM judges.

The source code and experimental materials used in this study are publicly available at https://github.com/iman-xosseini/injection_molding_study_sikdd2026.

Acknowledgements

This research was financially supported by the Slovenian Research and Innovation Agency (ARIS) via the Research Program Knowledge Technologies (grant P2-0103) and the Gravity project AI for Science (grant GC-0001); by the European Commission Marie Sklodowska Curie program through the Horizon Europe DN APRIORI project (GA 101073551). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. The European Union cannot be held responsible for them.

During the preparation of this manuscript, the authors used Microsoft Copilot, powered by GPT-5, to support grammar refinement, reference formatting, and overall manuscript organization. All AI-generated suggestions were carefully reviewed and revised by the authors, who take full responsibility for the final content of this publication.

References

[1] A. Bilal, D. Ebert, and B. Lin. 2025. LLMs for explainable AI: a comprehensive survey. arXiv preprint arXiv:2504.00125. (2025). arXiv: 2504.00125.

[2] J. Gu et al. 2025. A survey on LLM-as-a-judge. arXiv preprint arXiv:2411.15594. (2025). arXiv: 2411.15594.

[3] KAMP AI Data Center. 2026. AI dataset detail. Retrieved May 2026 from https://www.kamp-ai.kr/aidataDetail?AI_SEARCH=&page=1&DATASET_SEQ=4&EQUIP_SEL=&GUBUN_SEL=&FILE_TYPE_SEL=&WDATE_SEL=.

[4] A. Khan, R. Nahar, H. Chen, G. E. Constante Flores, and C. Li. 2025. Faultexplainer: leveraging large language models for interpretable fault detection and diagnosis. *Computers & Chemical Engineering*, 199, 109152. doi:10.1016/j.compchemeng.2025.109152.

[5] J. W. S. Lee. 2026. Injection molding analysis. Github repository. Retrieved May 2026 from https://github.com/johnwslee/injection_molding_analysis/tree/main.

[6] A. Presciuttini, A. Cantini, S. Cramer, M. Huber, D. Wolfschläger, R. H. Schmitt, and A. Portioli-Staudacher. 2025. An integrated framework for predictive quality in injection molding: combining explainable ai and time series analysis in a german industry case study. *IFAC-PapersOnLine*, 59, 10, 1677–1682. doi:10.1016/j.ifacol.2025.09.282.

[7] F. Tayalati, I. Boukrouh, A. Azmani, and M. Azmani. 2025. Interpreting injection molding quality defect using explainable artificial intelligence and analysis of variance. *Engineering Applications of Artificial Intelligence*, 147, 110362. doi:10.1016/j.engappai.2025.110362.

[8] S. Wen, F. Li, W. Zhuang, X. Pan, W. Yu, J. Bao, and X. Li. 2025. Leveraging large language models for human–machine collaborative troubleshooting of complex industrial equipment faults. *Advanced Engineering Informatics*.

Learning Molecular Representations from SMILES for Acute Toxicity Prediction

Simona Cholakova
University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies (UP FAMNIT)
Koper, Slovenia
simonacholakova04@gmail.com

Domen Vake
University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies (UP FAMNIT)
Koper, Slovenia
domen.vake@famnit.upr.si

Aleksandar Tošić
University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies (UP FAMNIT)
Koper, Slovenia
aleksandar.tosic@upr.si

Abstract

Predicting molecular toxicity from structure can reduce the cost and risk of experimental screening, but performance depends strongly on the molecular representation supplied to the predictor. We study whether embedding representations learned directly from SMILES strings, text-based descriptions of a molecule's chemical structure, can support acute toxicity prediction. A custom byte-pair tokenizer and six-layer Transformer encoder were pretrained by masked language modelling on five million unlabeled PubChem molecules, then fine-tuned independently on seven rat and mouse LD50 endpoints from TOXRIC. The resulting molecule-level embeddings were evaluated with the encoder's own regression head, and separately with an XGBoost regressor trained on the embeddings alone or concatenated with Morgan fingerprints. XGBoost achieved lower RMSE than the neural regression head on all seven endpoints, with reductions of 4.4–7.5%. Morgan fingerprints improved R^2 on six endpoints, although only modestly, and randomized SMILES augmentation improved validation loss in every task. The results show that self-supervised SMILES embeddings are useful features for toxicity modelling. Qualitative error analysis further suggests potential limitations for rare atoms, complex ring systems, and long sequences.

Keywords

acute toxicity, molecular representation, SMILES, self-supervised learning, Transformer, XGBoost

1 Introduction

Experimental toxicity testing is costly, slow, and potentially harmful. Quantitative structure-activity relationship (QSAR) models offer a complementary route: estimate a biological endpoint directly from molecular structure before experimental screening. Acute toxicity is commonly summarized by LD_{50}, the dose that is lethal to half of a test population. Because LD_{50} is a measurable endpoint available for multiple species and administration routes, it is also a useful benchmark for molecular machine learning.

The representation of a molecule largely determines what a QSAR model can learn. Classical descriptors and fingerprints encode predefined substructures efficiently, but may omit task-relevant interactions [10]. Learned representations instead optimize features from data. SMILES strings are especially attractive because they express molecular graphs as sequences and can therefore be processed by methods developed for language. Transformers [9] and masked language modelling [4] have already shown that large collections of unlabeled sequences can yield transferable representations; ChemBERTa extended this principle to molecular property prediction [3], pretraining on tens of millions of PubChem SMILES with a general-purpose subword tokenizer and evaluating its embeddings largely through an end-to-end prediction head. This leaves open whether a smaller, chemistry-aware encoder, evaluated separately from its own prediction head, produces embeddings that a downstream model can exploit as well or better, a question with direct practical value where labelled data and compute are limited. Large pretrained models do not automatically dominate fingerprint or graph baselines, and the value of their embeddings depends on both the downstream learner and the amount of labelled data.

We investigate a compact pipeline learned from raw molecular strings. A custom byte-pair encoding (BPE) tokenizer and Transformer encoder are pretrained on unlabeled PubChem SMILES, fine-tuned for seven acute toxicity endpoints, and then used as a fixed feature extractor for XGBoost. We evaluate three questions: (1) whether a tree ensemble can exploit the learned embedding better than the encoder's neural regression head; (2) whether Morgan fingerprints add complementary information; and (3) whether randomized SMILES augmentation helps when labelled datasets are limited. The study makes the following contributions:

- an end-to-end, self-supervised SMILES representation pipeline for LD_{50} regression;
- a controlled comparison of prediction heads and molecular feature combinations across seven TOXRIC endpoints; and
- an analysis of data augmentation, benchmark comparability, and systematic model failures.

2 Related Work

Molecular representations trade computational simplicity against structural detail. Fixed descriptors summarize selected physicochemical properties, while fingerprints encode the presence of local substructures in a fixed-length vector. Graph and three-dimensional methods preserve richer connectivity or geometry, but introduce additional modelling or conformer-generation costs. No representation is universally best: its value depends on the endpoint, available data, and downstream learner [10]. Our experiments therefore compare a learned sequential representation both alone and in combination with a widely used fingerprint.

Self-supervised molecular language models adapt techniques from natural language processing to chemical strings. ChemBERTa [3], for example, pretrained RoBERTa-style encoders on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.88

tens of millions of PubChem SMILES and demonstrated transferable molecular features. Its results also showed that scale alone does not guarantee superiority over graph or fingerprint baselines, but its evaluation left the learned representation and the end-to-end prediction head entangled, so it remains unclear how much of the reported performance reflects the embedding itself. We address this gap directly: our smaller, chemistry-aware encoder (a 512-token task-specific BPE vocabulary vs. ChemBERTa's general-purpose 52K-token vocabulary, pretrained on 5M vs. 77M PubChem molecules) is frozen after fine-tuning and its embeddings are handed to an independently tuned XGBoost model, isolating representation quality from the small regression head used during fine-tuning.

Acute toxicity prediction has been studied with descriptor-based QSAR and deep multi-task models. Jain et al. [5] shared information across species and exposure routes through consensus multi-task learning. TOXRIC later consolidated toxicity data and evaluated multiple model and descriptor combinations [11]. In contrast, we train one encoder per endpoint to measure representation behaviour without cross-task transfer, and use the published TOXRIC values only as contextual benchmarks because the validation protocols differ.

3 Methods

3.1 Data and tasks

Unlabeled pretraining data were drawn from PubChem [6]. We used five million canonicalized SMILES strings from the public PubChem-10M collection. No toxicity labels were exposed during this stage. Labelled data came from TOXRIC [11], a database that integrates toxicological measurements and published QSAR benchmarks. We selected all acute toxicity endpoints containing more than 2,000 compounds: oral, intravenous, and intraperitoneal LD_{50} for rats; and oral, subcutaneous, intravenous, and intraperitoneal LD_{50} for mice. Their sizes range from 2,323 to 35,299 compounds. Each endpoint was treated as an independent regression task using the log-transformed LD_{50} provided by TOXRIC.

For each endpoint, compounds were randomly divided into 70% training, 15% validation, and 15% held-out test sets with a fixed seed. Augmentation was applied only to the training split; validation and test molecules retained one canonical SMILES each.

3.2 Representation learning

Figure 1 summarizes the pipeline. Before BPE training [7], SMILES were split into chemically meaningful primitive units, including multicharacter atoms, bracketed atoms, bonds, branches, and ring closures. Frequent adjacent units were iteratively merged into a 512-token vocabulary, with reserved `[PAD]`, `[CLS]`, `[UNK]`, and `[MASK]` tokens. Sequences were prepended with `[CLS]` and padded or truncated to 128 tokens.

The encoder comprised six Transformer layers, together with a 512-dimensional hidden state, eight attention heads, a 1,024-dimensional feed-forward block, 0.15 dropout, and rotary positional embeddings [8]. During masked language model pretraining, 15% of non-special tokens were selected: 80% were replaced by `[MASK]`, 10% by a random token, and 10% left unchanged. Cross-entropy was computed only at selected positions. AdamW training used a learning rate of 10^{-4}, weight decay 0.01, batch size 256, and 15 epochs.

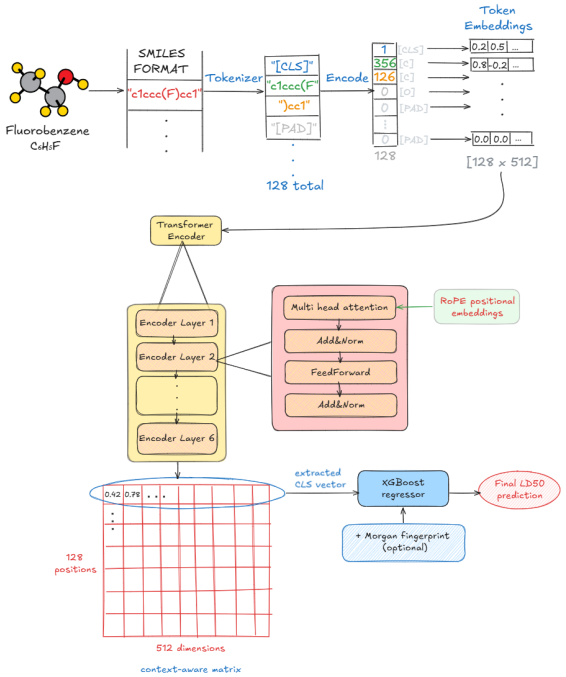


Figure 1: Pipeline from a SMILES string to the learned `[CLS]` embedding and final LD_{50} prediction. Morgan fingerprints are an optional additional feature.

3.3 Fine-tuning and prediction

The pretrained encoder was fine-tuned separately for each endpoint. A regression head mapped the 512-dimensional `[CLS]` state through a 256-unit ReLU layer and dropout to one LD_{50} value. Training minimized mean squared error with Adam, a learning rate of 10^{-4}, weight decay 5×10^{-5}, learning-rate reduction after three stagnant validation epochs, and early stopping with patience eight (maximum 50 epochs).

To increase invariance to SMILES syntax, we generated up to eight non-canonical forms per molecule using RDKit atom renumbering and `canonical=False` serialization. Identical fine-tuning runs with and without augmentation measured its effect.

After fine-tuning, the encoder was frozen; for each molecule, its canonical SMILES was passed through the encoder in evaluation mode with dropout disabled, and the final layer's [CLS] hidden state was taken as the 512-dimensional feature vector. XGBoost [2] was trained on (i) the 512 learned features and (ii) those features concatenated with a 2,048-bit radius-2 Morgan fingerprint. Training and validation features were combined; model selection used five-fold grid search over 500–2,000 estimators, depths 4–8, and learning rates 0.01–0.1. The held-out test set was evaluated once. We report root mean squared error (RMSE) and coefficient of determination (R^2).

3.4 Evaluation protocol

The neural head, XGBoost on `[CLS]`, and XGBoost on `[CLS]` plus Morgan features used identical endpoint splits. RMSE measures error on the scale of the log-transformed target and emphasizes large deviations; R^2 measures the explained test-set variance. Each result is from the untouched 15% test partition after model selection. The augmentation ablation instead compares the lowest validation mean squared error reached by otherwise identical fine-tuning runs. For external context, we compare with TOXRIC's

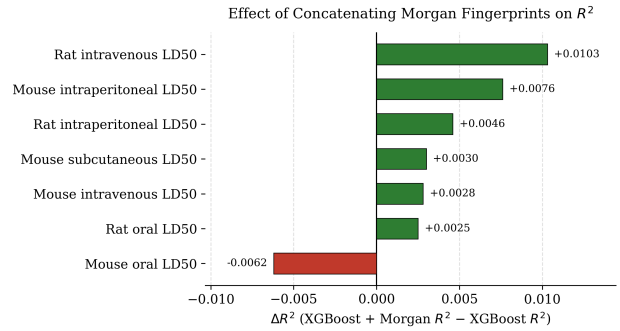


Figure 2: Change in R^2 after concatenating Morgan fingerprints with learned `[CLS]` embeddings.

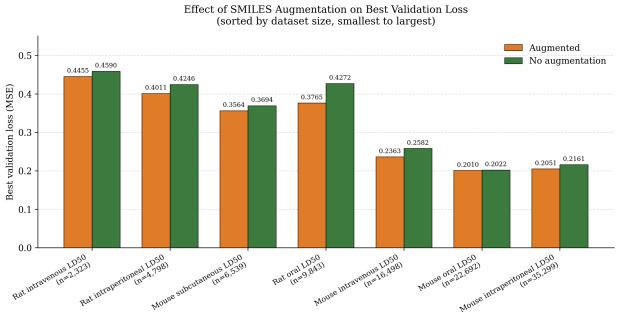


Figure 3: Best fine-tuning validation loss with and without randomized SMILES augmentation, ordered by endpoint size.

published results, but treat that comparison separately because TOXRIC reports five-fold cross-validation rather than our fixed holdout.

4 Results and Discussion

4.1 Prediction head and feature comparison

Table 1 reports held-out performance. XGBoost using learned embeddings outperformed the encoder's prediction head on all seven endpoints. Relative RMSE reductions ranged from 4.4% for mouse subcutaneous LD_{50} to 7.5% for rat oral LD_{50}. The result indicates that supervised fine-tuning produces an informative molecular embedding even when the small end-to-end regression head does not extract its full predictive value.

Adding Morgan fingerprints improved R^2 on six endpoints, but the gains were small (0.0025–0.0103). The largest increase occurred for rat intravenous LD_{50}, the smallest dataset; mouse oral LD_{50} was the sole negative case ($\Delta R^2 = -0.0062$). The largest gain occurred on the smallest endpoint, suggesting that classical fingerprints may be particularly useful as a complementary representation in lower-data settings. Learned and predefined representations capture overlapping, but not identical, structural information [10].

Dataset size alone did not determine difficulty. Rat intravenous had the highest RMSE and the fewest compounds, yet mouse oral had the lowest XGBoost R^2 despite being the second-largest endpoint. These results indicate that sample count alone does not explain task difficulty since differences in endpoint heterogeneity and chemical-space coverage may also contribute to it.

4.2 SMILES augmentation

Randomized SMILES augmentation reduced the best validation loss for every endpoint (Figure 3). Gains were generally larger with fewer labelled molecules. Rat oral improved by 11.9% (0.4272 to 0.3765) and mouse intravenous by 8.5% (0.2582 to 0.2363), whereas mouse oral improved by only 0.6% (0.2022 to 0.2010). Multiple valid serializations [1] encourage the encoder to model molecular structure rather than memorize one textual form, with greatest value when distinct labelled molecules are scarce.

4.3 Published benchmarks and failure modes

For context, we compared against TOXRIC's published XGBoost, random forest, support vector machine, and deep neural network results, which use PubChem fingerprints plus RDKit descriptors. Table 2 contrasts our best feature configuration per endpoint with the best of those four published models. Our approach achieved lower RMSE for mouse intravenous, rat oral, and rat intraperitoneal LD_{50}; the published models retained better RMSE on the other four tasks and better R^2 throughout. The R^2 gaps are particularly visible for the four mouse endpoints. This is only an approximate comparison: TOXRIC reports means from five-fold cross-validation, whereas our numbers use one held-out split. Direct claims of superiority would require identical folds and repeated fine-tuning.

Inspection of the three largest-error test molecules per endpoint revealed recurring limitations. Errors clustered around large fused-ring systems, compounds containing comparatively rare atoms such as silicon or iodine, and long SMILES strings (roughly 60–80 characters). The first two patterns suggest gaps in pretraining and endpoint coverage. Truncation was not the cause of the long-SMILES pattern in the present data: across all seven endpoints, every molecule fit within the 128-token limit (maximum 110 tokens, 0% truncated), so this difficulty likely stems from the greater structural complexity such strings tend to encode rather than from lost sequence information. However, this sequence length was sufficient only for the compounds in the current TOXRIC endpoints; larger or more structurally diverse molecular datasets could exceed it, so the 128-token limit should be revisited if the pipeline is extended to other chemical spaces. These observations motivate scaffold-aware evaluation, longer or structure-aware tokenization, and pretraining that better balances chemical space.

4.4 Limitations and validity

Three limitations constrain interpretation. First, random compound splits can place close structural analogues in both training and test sets; a scaffold-based split would more rigorously test extrapolation to unfamiliar chemistry. Second, results come from a single fixed train/validation/test split, so we cannot yet separate genuine representational gains from split-specific variance. We deliberately scope both a scaffold-aware split and repeated-seed or k-fold evaluation out of the present study, in favor of first establishing the XGBoost-versus-neural-head comparison and augmentation effects under controlled, identical conditions; we treat repeated, scaffold-aware cross-validation as the immediate next step, both to quantify this variance and to enable a fold-matched comparison with TOXRIC's published 5-fold results, rather than the current approximate one. Third, each endpoint was modelled independently, although related species and administration routes may contain transferable signal. The reported advantage of XGBoost over the neural head is controlled within

Table 1: Held-out test performance. "Head" is the fine-tuned neural regression head; "CLS" uses learned embeddings with XGBoost; "+Morgan" concatenates Morgan fingerprints. Best RMSE and R^2 within each endpoint are bold.

		Head		CLS		CLS + Morgan	
Endpoint	n	RMSE	R^2	RMSE	R^2	RMSE	R^2
Mouse oral	22,692	0.4741	0.3994	**0.4422**	**0.4775**	0.4448	0.4713
Mouse subcutaneous	6,539	0.6382	0.5098	0.6099	0.5523	**0.6078**	**0.5553**
Mouse intraperitoneal	35,299	0.4768	0.5277	0.4513	0.5768	**0.4473**	**0.5844**
Mouse intravenous	16,498	0.4811	0.5489	0.4490	0.6070	**0.4475**	**0.6098**
Rat oral	9,843	0.6227	0.5317	0.5761	0.5991	**0.5743**	**0.6016**
Rat intravenous	2,323	0.6936	0.5392	0.6481	0.5976	**0.6398**	**0.6079**
Rat intraperitoneal	4,798	0.6436	0.4538	0.6033	0.5200	**0.6004**	**0.5246**

Table 2: Contextual comparison with the best result among TOXRIC's four published models. Our result is the better of CLS and CLS + Morgan. Bold denotes the better value in each pair; protocols differ, so values are not a controlled ranking.

	RMSE (lower is better)		R^2 (higher is better)	
Endpoint	Ours	Best TOXRIC	Ours	Best TOXRIC
Mouse oral	0.4422	**0.4239**	0.4775	**0.6258**
Mouse subcutaneous	0.6078	**0.5658**	0.5553	**0.6212**
Mouse intraperitoneal	0.4473	**0.4407**	0.5844	**0.7047**
Mouse intravenous	**0.4475**	0.4557	0.6098	**0.7264**
Rat oral	**0.5743**	0.5759	0.6016	**0.6103**
Rat intravenous	0.6398	**0.6179**	0.6079	**0.6652**
Rat intraperitoneal	**0.6004**	0.6094	0.5246	**0.5481**

each split, but broader claims about representation quality require these additional evaluations. Finally, predicted LD_{50} values are screening aids, not substitutes for expert toxicological assessment; applicability-domain checks are especially important for the rare and complex molecules highlighted by the error analysis.

5 Conclusion

A Transformer trained by masked language modelling on unlabeled SMILES produces useful features for acute toxicity prediction. Across seven TOXRIC endpoints, XGBoost consistently exploited those features better than the encoder's neural regression head. Randomized SMILES augmentation reduced the best validation loss for every endpoint, while Morgan fingerprints provided small improvements on six endpoints and were most beneficial for the smallest dataset. The approach was competitive with published descriptor-based models, but did not dominate them across metrics and evaluation protocols. Future work should use scaffold-based, repeated cross-validation; multi-task learning across related species and administration routes [5]; and representations designed for long or underrepresented molecular structures. With the scaffold-aware and multi-task extensions outlined above, this pipeline could serve as a reusable foundation for broader molecular property prediction beyond acute toxicity. Code, the trained BPE tokenizer, and pretrained encoder weights (for both the CLS-only and CLS+Morgan configurations) are publicly available at https://github.com/simona-cholakova/molecular-representation-learning.

References

[1] Esben Jannik Bjerrum. 2017. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv preprint arXiv:1703.07076* (2017). doi:10.48550/arXiv.1703.07076

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794. doi:10.1145/2939672.2939785

[3] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv preprint arXiv:2010.09885* (2020). doi:10.48550/arXiv.2010.09885

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186. doi:10.18653/v1/N19-1423

[5] Sankalp Jain, Vishal B. Siramshetty, Vinicius M. Alves, Eugene N. Muratov, Nicole Kleinstreuer, Alexander Tropsha, Marc C. Nicklaus, Anton Simeonov, and Alexey V. Zakharov. 2021. Large-Scale Modeling of Multispecies Acute Toxicity End Points Using Consensus of Multitask Deep Learning Methods. *Journal of Chemical Information and Modeling* 61, 2 (2021), 653–663. doi:10.1021/acs.jcim.0c01174

[6] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. 2023. PubChem 2023 Update. *Nucleic Acids Research* 51, D1 (2023), D1373–D1380. doi:10.1093/nar/gkac956

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 1715–1725. doi:10.18653/v1/P16-1162

[8] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing* 568 (2024), 127063. doi:10.1016/j.neucom.2023.127063

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[10] Daniel S. Wigh, Jonathan M. Goodman, and Alexei A. Lapkin. 2022. A Review of Molecular Representation in the Age of Machine Learning. *WIREs Computational Molecular Science* 12, 5 (2022), e1603. doi:10.1002/wcms.1603

[11] Lianlian Wu, Bowei Yan, Junshan Han, Ruijiang Li, Jian Xiao, Song He, and Xiaochen Bo. 2023. TOXRIC: A Comprehensive Database of Toxicological Data and Benchmarks. *Nucleic Acids Research* 51, D1 (2023), D1432–D1445. doi:10.1093/nar/gkac1074

Logical Anomaly Detection Using Segmentation Transformers

Yunfei Liang
yl42484@student.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Matic Fučka
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Danijel Skočaj
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Abstract

Industrial visual inspection aims to detect and localize defects in manufactured products. This task is commonly addressed using anomaly detection methods that model the appearance of defect-free samples. Such methods perform well on structural anomalies, such as dents and scratches, but they often struggle with logical anomalies. For logical anomalies, individual components appear normal in isolation but their composition may violate constraints of the category (when a component is missing, duplicated, or misplaced). To detect logical anomalies, we propose a two-branch method that uses a Mask2Former-based component segmenter to model both local appearance and global composition. The structural branch extracts local information from a frozen backbone's patch-level features and scores each patch against a coreset-reduced memory bank of defect-free patches. The compositional branch derives a descriptor from the predicted segmentation masks by pooling the same frozen features inside each component region, and compares each component against defect-free components of the same class in a class-partitioned memory bank. The regions establish correspondence with normality through their given class labels, without open-world object detection or inference-time bipartite matching. During inference, anomaly scores are obtained by comparing the local and compositional representations of a query image with those of defect-free training samples, and the two resulting maps are fused into a single prediction. On the MVTec LOCO dataset, the proposed method achieves an image-level AUROC of 95.7% using hand-labeled part maps released with PSAD and 92.9% using automatically generated composition maps released with SALAD, two component-based anomaly detection methods.

Keywords

anomaly detection, logical anomaly detection, semantic segmentation, visual inspection, machine vision

1 Introduction

Industrial anomaly detection learns the appearance of defect-free products and flags deviations at test time, a setting motivated by the fact that anomalous examples are rare and that the defects encountered in production are often not known in advance [3]. Most work has focused on *structural* anomalies such as scratches, dents, contaminations, or other locally visible defects. MVTec LOCO [2] introduced a more difficult setting, *logical* anomalies, where individual components may appear normal in isolation while their composition is incorrect: a component may be missing, duplicated, extra, or placed in an invalid configuration.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.115

The difficulty is that the evidence for a logical anomaly often does not reside in any single abnormal-looking region. A missing component can expose pixels that are themselves perfectly normal for the object category, leaving nothing for a purely appearance-based detector to react to. A local nearest-neighbor detector can compare what is *present* against normal appearance, but it has no direct mechanism for representing what should have been present and is not. Logical anomaly detection therefore requires a representation of expected composition in addition to local appearance.

Several families of methods address this problem. Local–global approaches add a global branch to a local anomaly detector, using image-wide reconstruction or distillation error as an indirect proxy for compositional violations; GCAD, introduced with MVTec LOCO [2], and EfficientAD [1] follow this pattern. Explicit semantic-component approaches instead segment the image into component classes and summarize the resulting composition. ComAD [11], PSAD [10], SALAD [7], and CSAD [9] belong to this family. In our experiments, PSAD supplies manually annotated part maps, while SALAD supplies automatically generated composition maps for training the segmenter. A third line of work represents an image as a set or collection of elements. SINBAD [5] models set-level feature statistics, while ObjectCore [6] and SAM-LAD [12] reason over explicit object representations. ObjectCore is particularly close to our method: it retrieves a normal object set and aligns the query objects to it with bipartite matching, so missing, extra, and visually altered objects become unmatched or high-cost entries.

Our method keeps the object-representation idea but changes the correspondence mechanism. Because each predicted component has a semantic class label, a test component of class c can be compared directly against normal components of the same class in a class-partitioned memory bank. This removes the need for an open-world detector, a separate mask-refinement stage, and inference-time Hungarian matching. The cost of this simplification is that the representation is class-level rather than instance-level: multiple objects of the same class are merged into one semantic region. To compensate for the corresponding loss of sensitivity to small local defects, we retain a separate PatchCore-style structural branch and fuse the two anomaly maps.

Our contributions are: (i) a logical anomaly detector that uses semantic class labels to establish correspondence with normal components; (ii) a complementary structural branch and calibrated map fusion; and (iii) ablations examining the effects of scoring features, segmentation quality, and component-label granularity.

2 Method

The detector has two branches built around a shared frozen anomaly-scoring representation. The structural branch compares local patch descriptors against a coreset-reduced memory of normal patches. The logical branch first decomposes the image into semantic components, pools the frozen scorer features inside

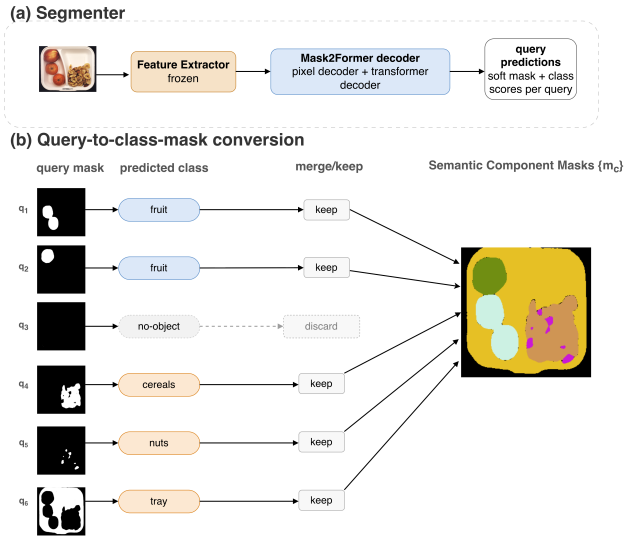


Figure 1: Conversion of Mask2Former query predictions to class masks. Active queries are assigned to their predicted semantic classes, and masks of the same class are merged by element-wise maximum to produce one soft mask m_c per detected class.

each component region, and compares the resulting descriptors against class-specific normal memories. Only the segmentation network is trained; the feature extractor used for anomaly scoring stays frozen, and both anomaly memory banks are constructed directly from normal training images.

Semantic component extraction. We use a Mask2Former-style semantic segmenter [4]. Its feature backbone is frozen, while the segmentation-specific layers are trained from component annotations. For each input image the decoder produces a fixed set of query predictions. Each query outputs a class distribution, including a no-object class, together with a soft spatial mask. Queries whose no-object probability is below 0.5 are retained and assigned to their most likely semantic class. Because several queries can predict the same class, their soft masks are merged by an element-wise maximum,

$$m_c(p) = \max_{q \in \mathcal{A}_c} m_q(p),$$

where $\mathcal{A}_c$ is the set of active queries assigned to class c. We discard merged masks whose mean soft-mask value is below 5×10^{-4} and exclude background from the component classes. The result is one soft mask m_c for each component class detected in the image (Figure 1).

Component representation. The semantic masks specify the location of each component class but do not describe its appearance. We therefore extract a feature map f from a frozen anomaly-scoring backbone and use each soft mask m_c to pool the scorer features inside the corresponding region:

$$\tilde{o}_c = \frac{1}{\sum_p m_c(p)} \sum_p m_c(p)\, f(p), \qquad o_c = \frac{\tilde{o}_c}{\|\tilde{o}_c\|_2}. \tag{1}$$

The soft mask acts as a spatial weighting function, so pixels with high mask confidence contribute more strongly to the descriptor. We additionally record the relative area $a_c = \frac{1}{HW} \sum_p m_c(p)$, which captures how much of the image is occupied by the component. An image is thus represented as the set $O = \{(o_c, m_c, a_c)\}_{c \in C}$ of detected semantic components.

Logical memory and scoring. For each anomaly-free training image, its components are added to a class-partitioned logical memory. For each class c, the bank contains the unit-normalized descriptors of all normal components of that class, their corresponding relative areas, and a mean soft mask $\bar{m}_c$. The mean mask is computed from the predicted soft masks in training images where the class is detected: $\bar{m}_c = \frac{1}{n_c} \sum_{i:\, c \in C_i} m_c^{(i)}$, where C_i denotes the classes detected in training image i and n_c is the number of training images containing a detection of class c. Images where the class is absent are excluded from this average. All masks are represented on the same 256×256 grid.

We also record the fraction of normal training images in which each class is detected. With N training images for the product category, a class is considered expected when $n_c/N \geq 0.5$. An expected class that is not detected in the test image is treated as missing.

For a test component of class c represented in the normal memory, we retrieve its nearest normal descriptor o_s^c using cosine distance alone, together with the relative area a_s^c of that same component. The semantic label therefore supplies the correspondence that ObjectCore must recover with set matching. The component cost combines the appearance difference with a penalty for the relative-area difference between the test component and its selected normal neighbor:

$$d_c = 1 - \langle o_t^c, o_s^c \rangle + \gamma \frac{|a_t^c - a_s^c|}{\max(a_t^c, a_s^c, \varepsilon)}. \tag{2}$$

The weight $\gamma = 0.25$ controls the contribution of the area penalty, while $\varepsilon = 10^{-6}$ keeps the denominator positive.

Let $\mathcal{M}$ denote classes present in both the test image and the normal memory, $\mathcal{E}$ classes detected in the test image but never detected in training, and $\mathcal{S}$ expected normal classes that are missing from the test image. The logical anomaly map is

$$A_L = \sum_{c \in \mathcal{M}} m_t^c\, d_c + \sum_{c \in \mathcal{E}} m_t^c + \sum_{c \in \mathcal{S}} \bar{m}_c. \tag{3}$$

Matched components therefore contribute their test masks weighted by their matching costs. Extra components contribute their own masks directly. A missing component has no test mask, so its mean normal mask is used to localize the region where the component is normally expected.

Structural branch and fusion. The logical descriptor is intentionally coarse: a scratch occupying only a small fraction of a component may barely change its mask-pooled appearance and will not change the component vocabulary. We therefore retain a separate PatchCore-style structural branch [13]. Each spatial feature of the same frozen scorer is treated as a patch descriptor, with no local aggregation in the final $p = 1$ configuration. Normal patch descriptors are reduced to a 100,000-entry bank with greedy k-center coreset subsampling. At test time each patch is scored by squared Euclidean distance to its nearest normal patch descriptor, producing the structural anomaly map A_S.

Because the two branches have different score distributions, each map is calibrated using the 99.5th percentile estimated from anomaly-free validation images. The calibrated maps are combined by a weighted sum, followed by Gaussian smoothing:

$$A = G_\sigma * \left(w_S \frac{A_S}{s_S} + w_L \frac{A_L}{s_L} \right),$$

Table 1: Combined image-level AUROC (%) on MVTec LOCO for the strongest and closest baselines. Rows above the divider use automatically generated component maps or none; rows below use PSAD's hand-labeled maps (SALAD†: SALAD trained with them). B.box = breakfast box; J.bottle = juice bottle; Pins = pushpins; S.bag = screw bag; S.conn. = splicing connectors.

Method	B.box	J.bottle	Pins	S.bag	S.conn.	Avg.
ComAD+PC [11]	86.4	96.6	93.4	80.2	94.1	90.1
SAM-LAD [12]	91.0	97.6	88.2	86.6	90.0	90.7
CSAD [9]	92.8	95.3	98.7	96.6	93.5	95.3
SALAD [7]	89.3	99.7	99.4	95.0	97.3	96.1
Proposed method (SALAD maps)	96.9	95.9	88.5	91.9	91.2	92.9
PSAD [10]	92.5	98.7	94.9	97.5	90.6	94.9
SALAD† [7]	94.2	99.3	99.1	96.6	97.0	97.2
Proposed method (PSAD maps)	96.4	98.5	94.5	96.0	93.3	95.7

Table 2: Branch isolation at the reference configuration.

Maps	Configuration	$\text{AUROC}_{\text{log}}$	$\text{AUROC}_{\text{str}}$	Mean	sPRO
–	structural only	0.7338	0.9740	0.8539	0.7082
PSAD	logical only	0.9399	0.7498	0.8448	0.5517
PSAD	fused	**0.9588**	0.9560	**0.9574**	**0.7693**
SALAD	logical only	0.8703	0.7579	0.8141	0.2986
SALAD	fused	**0.8793**	**0.9783**	**0.9288**	0.6207

Table 3: Selected one-factor-at-a-time ablations. Δ is the change in mean AUROC from the corresponding reference configuration.

Maps	Change	Mean AUROC	Δ
PSAD	scorer RADIO-L → WRN50	0.8711	-0.0863
PSAD	greedy → random coreset	0.9458	-0.0115
PSAD	γ : 0.25 → 1.0	0.9394	-0.0180
PSAD	MSDef → FPN	0.9545	-0.0029
SALAD	scorer RADIO-L → WRN50	0.8816	-0.0472
SALAD	greedy → random coreset	0.9200	-0.0087

and the image-level anomaly score is the maximum value of the fused map. The reference configuration uses $w_S = w_L = 1$ and $\sigma = 4$.

3 Experiments

Dataset and evaluation. All experiments use the five product categories of MVTec LOCO [2]. The training and validation splits contain only anomaly-free images, while the test split contains normal images together with structural and logical anomalies. We use two released component-annotation sources. SALAD [7] provides automatically generated composition maps with 18 relatively coarse classes, whereas PSAD [10] provides 53 hand-labeled part classes, some of which encode position. The same anomaly pipeline is used with both sources; no new component annotations are introduced for these experiments.

We report image-level AUROC separately for logical and structural anomaly subsets, their unweighted mean, and saturated per-region overlap (sPRO) at FPR≤0.05, using the official MVTec LOCO evaluator. The segmenter is trained separately for each category for 100 epochs with AdamW using a frozen RADIO v2.5-B backbone [8] and a multi-scale deformable-attention pixel decoder. Anomaly scoring uses a frozen RADIO v2.5-L at 1024 px, squared L_2 distance in the structural branch, no local aggregation, a greedy 100,000-entry structural coreset, area weight $\gamma = 0.25$, and the fusion settings above.

Comparison to prior work. Table 1 summarizes the closest and strongest baselines. With PSAD component maps, the fused detector reaches 95.7% combined AUROC, above PSAD's own 94.9% and within 1.5 percentage points of the supervised SALAD† result of 97.2%. With automatically generated SALAD maps, the same pipeline reaches 92.9%, below SALAD's 96.1% result. ObjectCore, the closest object-representation reference, reports 90.8% in its full-shot setting; both of our configurations exceed it. The comparison therefore shows that semantic class correspondence can support competitive logical anomaly detection, although the result depends strongly on the component vocabulary used to train the segmenter.

Branch contributions. The two branches behave as intended (Table 2). The structural branch reaches 0.974 AUROC on structural anomalies but only 0.734 on logical anomalies. The logical branch shows the opposite pattern, reaching 0.940 logical AUROC with PSAD maps but only 0.750 structural AUROC. Fusing the two branches increases mean AUROC by more than 0.07 over either isolated branch for both annotation sources.

Main ablations. Before interpreting small changes, we estimate the variation caused by segmenter training randomness across three seeds in earlier PSAD experiments. The standard deviation in mean AUROC ranges from 0.0018 to 0.0054. We use these runs as a practical reference when interpreting small differences and treat changes of around 0.005 or less cautiously.

Among the ablations in Table 3, the frozen anomaly-scoring backbone has by far the largest effect. Replacing RADIO v2.5-L with WideResNet-50-2 reduces mean AUROC by 0.086 with PSAD maps and 0.047 with SALAD maps. By comparison, replacing the multi-scale deformable-attention decoder with an FPN or replacing the segmenter backbone produces much smaller changes in mean AUROC. This indicates that, in these experiments, the pretrained representation used for anomaly scoring matters much more than the architecture used to generate the component masks.

Greedy coreset selection gives a small but consistent gain over random subsampling. The relative-area term matters mainly with the finer PSAD vocabulary: increasing γ from 0.25 to 1.0 reduces mean AUROC by 0.018 on PSAD, while the SALAD result is essentially unchanged. These differences are consistent with the role of area as an indirect cue for count or occupancy when a semantic class merges multiple instances.

Segmentation quality is not the main bottleneck. We additionally test whether improving semantic segmentation improves anomaly detection. Full backbone fine-tuning, late-stage fine-tuning, and LoRA adaptation increase strict segmentation mIoU by roughly 0.006–0.011, yet none improves mean AUROC by more than 0.005 once configurations are compared at the same area-term setting. One possible explanation is that the anomaly detector never consumes the final hard segmentation map directly. Instead, the predicted soft masks are used to pool features from a separate frozen scorer, so moderate changes in hard pixel agreement can leave the pooled component descriptors nearly unchanged.

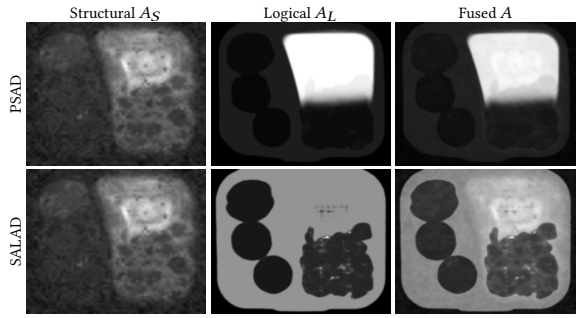


Figure 2: Breakfast box with granola missing from the upper-right compartment. Rows show results using PSAD and SALAD component maps. The structural map A_S is identical in both rows because the structural branch does not use component maps. Each panel is normalized to its own maximum. With PSAD maps, the missing granola component is localized using its mean normal mask $\bar{m}_c$; with SALAD maps, the logical response is over the whole box.

Annotation granularity and position. The annotation source has a much larger effect on the logical branch. At the reference setting, PSAD reaches 0.9588 logical AUROC whereas SALAD reaches 0.8793. The gap is largest for `pushpins` and `screw_bag`. The finer PSAD vocabulary contains more part classes and, for some categories, position-coded labels. Consequently, some arrangement violations that are difficult to express with a coarse semantic class become explicit missing- or extra-class events in the class-partitioned representation. Figure 2 illustrates this on a breakfast box with the granola missing: with PSAD maps the absent class is detected as missing and A_L concentrates on the empty compartment, whereas with SALAD maps the logical response is spread over the whole box.

Because the class-level representation discards some spatial arrangement information, we also tested an explicit class-conditional position term based on the centroid of each predicted component. This extension does not help: increasing its weight decreases mean AUROC on both annotation sources, and at $\gamma_{\text{pos}} = 1$ the mean is roughly 0.075 lower for PSAD and 0.062 lower for SALAD than with the term disabled. The penalty raises scores on normal images as well, reducing the separation between normal and anomalous examples. We therefore use $\gamma_{\text{pos}} = 0$ in the reference model.

Fusion sensitivity. The optimal branch balance depends on the annotation source. With PSAD maps, reducing the logical weight to $w_L = 0.5$ increases mean AUROC from 0.9574 to 0.9650 and also improves sPRO, whereas the same change reduces the SALAD result. We keep equal weights in the reference configuration because MVTec LOCO provides no anomalous validation set on which to choose the branch weights without using the test data.

4 Conclusion

We presented a two-branch anomaly detector in which semantic class labels provide correspondence between a test image and normality. The logical branch uses mask-pooled component descriptors and class-partitioned memories, while a PatchCore-style structural branch captures local appearance faults.

The main limitation is that one semantic mask is retained per class, so multiple same-class instances are not represented independently. Intra-class count is therefore captured only indirectly through pooled appearance and relative area. A natural extension is to convert the semantic segmenter to panoptic output so that individual instances remain separate. The scorer experiments also suggest that the two branches may prefer different frozen feature extractors, making branch-specific scorers another useful direction for future work.

We train the segmenter using component maps released with PSAD and SALAD and create no additional annotations. Applying the method to new categories requires manual labeling effort or an automatic component-annotation pipeline. Our experiments cover only MVTec LOCO, and we do not evaluate runtime. The code is not currently publicly available.

Acknowledgements

This work was in part supported by the ARIS research projects MUXAD (J2-60055) and AI4Science (GC-0001), research programme P2-0214 and the supercomputing network SLING (ARNES, EuroHPC Vega).

References

[1] Kilian Batzner, Lars Heckler, and Rebecca König. 2024. EfficientAD: accurate visual anomaly detection at millisecond-level latencies. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 127–137.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. 2022. Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130, (Apr. 2022).

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. 2019. MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9592–9600.

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1280–1289.

[5] Niv Cohen, Issar Tzachor, and Yedid Hoshen. 2024. Set features for anomaly detection. *Transactions on Machine Learning Research*.

[6] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. 2026. ObjectCore – efficient few-shot logical anomaly detection using object representations. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 3857–3867.

[7] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. 2025. SALAD – semantics-aware logical anomaly detection. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, 21843–21852.

[8] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. 2025. RADIOv2.5: improved baselines for agglomerative vision foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22487–22497.

[9] Yu-Hsuan Hsieh and Shang-Hong Lai. 2024. CSAD: unsupervised component segmentation for logical anomaly detection. *arXiv preprint arXiv:2408.15628*.

[10] Soopil Kim, Sion An, Philip Chikontwe, Myeongkyun Kang, Ehsan Adeli, Kilian M Pohl, and Sang Hyun Park. 2024. Few shot part segmentation reveals compositional logic for industrial anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence* number 8. Vol. 38, 8591–8599.

[11] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Xiao Jin, Liuyi Jin, and Zhuo Zhao. 2023. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics*, 58, 102161.

[12] Yun Peng, Xiao Lin, Nachuan Ma, Jiayuan Du, Chuangwei Liu, Chengju Liu, and Qijun Chen. 2025. SAM-LAD: Segment Anything Model meets zero-shot logic anomaly detection. *Knowledge-Based Systems*, 314, 113176.

[13] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14298–14308.

Posterior Contraction Rate and Roadmap to Semiparametric Bernstein-von Mises Theorem for Bayesian Causal Forests

Andrej Srakar
International Research Centre on Artificial Intelligence under the auspices of UNESCO and Department for Artificial Intelligence
Jožef Stefan Institute, Ljubljana, Slovenia
andrej.srakar@ijs.si

Abstract

Bayesian causal forest is a presently widely used and discussed approach in Bayesian causal inference. It is a machine learning model used to find how much a treatment or intervention affects different units using observational data. It consists of a Bayesian additive regression trees outcome regression with an estimated propensity score as predictor, which leads to a product-BART Bayesian framework. We derive posterior contraction rate for the model, as well as most of proofs for the corresponding BCF semiparametric Bernstein-von Mises theorem. We conclude by discussing some open research directions.

Keywords

Bayesian causal forest, BART, posterior contraction rate, semiparametric BvM theorem, product-BART, product probability spaces, asymptotics, weak learners, artificial intelligence

1 Introduction

Bayesian causal inference is an interesting and mportant area of future development of both Bayesian and causal methods and probability theory in general. One of most common Bayesian priors for causal inference remains Bayesian additive regression trees (BART, see Chipman et al., 2010). Theoretical properties of BART have been subject of important recent research by Ročkova (Ročkova and van der Pas, 2020; Jeong and Ročkova, 2023). In her recent contribution on the topic Jeong and Ročkova propose a new class of sparse piecewise heterogeneous anisotropic Hölder functions and derive their minimax optimality properties, which lead to BART posterior contraction rate. But, they do not prove a Bernstein-von Mises theorem for BART. We extend their work to Bayesian Causal Forests (BCF), proposed by Hahn, Murray and Carvalho (Hahn et al., 2020).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2025 Copyright held by the owner/author(s).
DOI: 10.5281/zenodo.22911909

BCF is a modular Bayesian approach, based on regression decomposition of Robinson into prognostic and treatment effect part of the outcome regression specification, which include propensity score parameter. In our article we extend theoretical derivations of Ročkova and van der Pas, and Jeong and Ročkova to accomodate for the modular and product-space nature of BCF. Posterior contraction rate and most proofs needed for semiparametric Bernstein-von Mises theorem for the BCF approach are derived. Our article delves into an important area of asymptotics for Bayesian nonparametric priors for causality, and we discuss also extensions to other BNP approaches for causality outside of the BART framework.

2 Derivation of posterior contraction rate for BCF

The BART prior is the hierarchical prior used in Bayesian Additive Regression Trees to regularize the ensemble of trees. It places priors on the tree structures; the terminal node (i.e. leaf) parameters; and the error variance.

Let the regression function be represented as a sum of m trees: $y_i = f(x_i) + \varepsilon_i$,where $f(x) = \sum_{j=1}^{m} g(x; T_j, M_j)$ $T_j$is the structure of the j-th tree, $M_j = \mu_{j1}, \ldots, \mu_{jb_j}$ are the terminal-node values, $g(x; T_j, M_j)$ maps x to the leaf value corresponding to its terminal node, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

The prior encourages small trees. For a node at depth d, the probability it is split is $P(\text{split at depth } d) = \alpha(1+d)^{-\beta}$, where typically $\alpha = 0.95, \beta = 2$. Given that a node splits, the splitting variable is chosen uniformly and the split point is chosen uniformly among available cutpoints. Thus, $p(T) = \prod_{\eta \in \mathcal{I}(T)} \alpha(1+d_\eta)^{-\beta} \prod_{\eta \in \mathcal{L}(T)} 1 - \alpha(1+d_\eta)^{-\beta} \times p(\text{split rules})$ where $\mathcal{I}(T)$ are internal nodes and $\mathcal{L}(T)$ are terminal nodes. If tree J has b_j terminal nodes, $M_j = \mu_{j1}, \ldots, \mu_{jb_j}$, then $\mu_{jk} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \tau^2)$.

The full BART prior can be summarized in five parts:

1) $y_i \mid f, \sigma^2 \sim \mathcal{N}(f(x_i), \sigma^2)$

2) $f(x) = \sum_{j=1}^{m} g(x; T_j, M_j)$

3) $T_j \sim p(T)$

4) $\mu_{jk} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \tau^2)$

5) $\sigma^2 \sim \text{Inverse-Gamma}(\frac{\nu}{2}, \frac{\nu\lambda}{2})$

The defining characteristic of the BART prior is the combination of a depth-penalizing prior on tree structures, Gaussian shrinkage on leaf values scaled by $1/\sqrt{m}$, and a prior on the residual variance. This induces strong prior toward many shallow, weak trees, with the sum flexibly approximating complex nonlinear regression functions.

Theorem of Rockova and van der Pas proves posterior contraction for BART. Assume $Y_i = f_0(x_i) + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)$, where $x_i \in [0,1]^p$, $f_0 \in C^{\alpha}([0,1]^p)$ and only q variables are relevant. Then under a suitably regularized BART prior, $\Pi(|f - f_0|_n > M\varepsilon_n \mid Y) \to 0$, where $\varepsilon_n = n^{-\alpha/(2\alpha+q)}(\log n)^c$. The proof is an application of the general Bayesian posterior contraction theorem. One must verify three ingredients (see van der Vaart and van Zanten, 2008): existence of exponentially consistent tests; sufficient prior mass near f_0; and exponentially small prior mass outside an appropriate sieve.

Bayesian causal forest (BCF) model developed in 2020 in a Bayesian Analysis article by Hahn, Murray and Carvalho is an extension of BART by including estimate of the propensity score in the outcome regression equation. It writes $Y_i = \mu(X_i, \hat{e}_i) + \tau(X_i)W_i + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)$, Where $\mu \sim \Pi_\mu, \tau \sim \Pi_\tau$ are independent BART priors. The target is $\psi_0 = E_{P_X}[\tau_0(X)]$.

BCF are not a single BART prior but a joint prior on two regression functions with different roles: $Y_i = \mu(X_i, \hat{\pi}_i) + Z_i\tau(X_i) + \varepsilon_i$, where $\mu(\cdot)$i is the prognostic function, $\tau(\cdot)$ is the treatment effect function, each is given an independent BART prior and $\hat{\pi}$ is an estimated propensity score included as a covariate. Consequently, a Bernstein–von Mises (BvM) theorem for BCF is fundamentally a semiparametric BvM problem with nuisance functions, rather than simply a BvM theorem for BART. Semiparametric Bernstein-von Mises theorems have been proposed in research works from 2012-2014 by Ismael Castillo and Richard Nickl (see Castillo and Nickl, 2014) who proposed two broad solutions to develop them for general nonparmetric stituations.

Coherent proof strategy to develop a semiparametric Bernstein-von Mises theorem for BCF consists of six parts: 1) Extend the Rocková–van der Pas posterior contraction theory to the product BART prior $\Pi_\mu \otimes \Pi_\tau$; 2) Establish LAN for the joint likelihood in the directions (h_μ, h_τ); 3) Characterize the nuisance tangent space associated with μ and derive the efficient influence function for the target functional of τ; 4) Prove a no-bias condition using the posterior contraction rates for both BART components; 5) Linearize the target functional and show the remainder is $o_p(n^{-1/2})$; 6) Apply a semiparametric BvM theorem (in the spirit of results by van der Vaart, Rousseau, and Kleijn, see van der Vaart and van Zanten, 2008; Rousseau, 2010; Kleijn, 2021) to conclude asymptotic normality of the posterior for the treatment effect functional.

The main idea for the proof is that you are not proving a new contraction theorem for one BART prior, but for the product prior $\Pi = \Pi_\mu \otimes \Pi_\tau$, on the product space $\mathcal{F} = \mathcal{F}_\mu \times \mathcal{F}_\tau$. The proof follows the same high-level structure as the single-BART case, but ingredients must be lifted to a product of function spaces.

Our main theorem is (Product BART contraction):

Assume $Y_i = \mu_0(X_i) + Z_i\tau_0(X_i) + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$, where$Z_i \in 0, 1$. Suppose $\mu_0 \in C^{\alpha_\mu}, \tau_0 \in C^{\alpha_\tau}$, and $\Pi = \Pi_\mu \otimes \Pi_\tau$ is the independent product of two regularized BART priors. Assume overlap $0 < c < P(Z = 1|X = x) < 1 - c$ for some c>0. Then $\Pi(d((\mu, \tau), (\mu_0, \tau_0)) > M\varepsilon_n \mid Y) \to 0$, where $d^2 = |\mu - \mu_0|n^2 + |\tau - \tau_0|n^2$, and $\varepsilon_n^2 = \varepsilon\mu, n^2 + \varepsilon\tau, n^2$ with $\varepsilon_{\mu,n} = n^{-\alpha_\mu/(2\alpha_\mu+d)}(\log n)^{c_\mu}$, and $\varepsilon_{\tau,n} = n^{-\alpha_\tau/(2\alpha_\tau+d)}(\log n)^{c_\tau}$. Consequently, $\varepsilon_n = O(\max(\varepsilon_{\mu,n}, \varepsilon_{\tau,n}))$.

Complete proof exceeds length of this article and is accessible as a separate document. Above theorem proves that posterior contraction rate for BCF is of order of maximum of standard errors of treatment and intervention parts of the BCF regression equation. Additionally to the above, information operator coercivity proof is a special part not usual for such theorems in Bayesian asymptotics.

Strategy to derive the proof can be described shortly as follows. Firstly, we prove product approximation, product prior mass, product entropy, product sieve, and joint testing theorems which extend BART to a product-BART situation. This included specially considering and addressing problems of Kullback-Leibler (KL) likelihood geometry, approximation, prior concentration, entropy, sieve probability, testing, KL neighbourhoods, and metric equivalence. For the latter, we proved coercivity estimate by proving that information matrix is uniformly positive definite, and that its eigenvalues are bounded below.

3 Path to semiparametric BvM theorem for BCF

At this point, complete semiparametric Bernstein-von Mises theorem remains subject to our next work, as some part demand additional proofs and verifications. But we are able to prove some of its important parts.

Theorem (Conditional LAN Expansion over Product-BART Neighborhoods):

Consider $Y_i = \mu(X_i) + Z_i\tau(X_i) + \varepsilon_i, \varepsilon_i \overset{iid}{(0, \sigma^2)}$, with true functions (μ_0, τ_0). Define $f(x, z) = \mu(x) + z\tau(x)$, and $f_0(x, z) = \mu_0(x) + z\tau_0(x)$. Suppose $|f - f_0|_n \leq M\varepsilon_n$, where $\varepsilon_n = n^{-\alpha/(2\alpha+p)}(\log n)^k$. Assume additionally product-BART posterior contracts uniformly over this neighborhood; tree depth is uniformly bounded by $D_n = o(\log n)$; leaf values satisfy $\sup_x |f(x)| < C$; the empirical design satisfies $\frac{1}{n}X^\top X \to \Sigma$.

Then $\sup_{|f-f_0|_n \leq M\varepsilon_n} \ell_n(f) - \ell_n(f_0)\sqrt{n}, W_n(h) + \frac{1}{2}n|h|n^2 = o_P(1)$, where $h = f - f_0$ and $W_n(h) = \frac{1}{\sigma^2\sqrt{n}} \sum i = 1^n \varepsilon_i h(X_i, Z_i)$.

Theorem (Prior Stability Under Least-Favorable Perturbations):

Let $f = (\mu, \tau)$ have an independent Product-BART prior $\Pi = \Pi_\mu \otimes \Pi_\tau$. Let $\eta = (\eta_\mu, \eta_\tau)$ denote the least-favorable direction associated with the efficient influence function. Define the shifted parameter $f_t = f + \frac{t}{\sqrt{n}}\eta t \in \mathbb{R}$. Suppose: posterior contraction holds; tree depth satisfies $D_n = o(\log n)$, leaf parameters satisfy $|\omega_j| \leq C$; every Product-BART density is locally Lipschitz; perturbations leave tree topology unchanged with probability $1 - o(1)$.

Then $\frac{d\Pi(f_t)}{d\Pi(f)}1 + o(1)$ uniformly over $|f - f_0|_n \leq M\varepsilon_n$. This is the no-bias condition.

In order to finalize also the proof of Bernstein-von Mises theorem, one needs the following:

- An inverse approximation inequality relating empirical L_2 and L_∞ errors for Product-BART function classes.

- High-probability bounds on the maximum diameter of random Product-BART partition cells.

- A margin theorem for split stability, showing that scale perturbations do not alter selected partitions.

- A proof that the effective number of leaves grows slow enough under the Product-BART prior.

- Uniform versions of these bounds over the posterior contraction neighborhood.

This forms a still missing final roadmap toward a complete semiparametric Bernstein–von Mises theorem for Bayesian Causal Forests.

4 Discussion and open problems

Our article derives posterior contraction rate for Bayesian causal forests which has not been derived thus far. It also derives most parts in order to prove the semiparametric Bernstein-von Mises theorem for BCF. Analysis takes part using product-BART structure of the Bayesian causal forests regression problem.

Our proof derivation opens several interesting research directions of future work. Firstly, one is now able to prove semiparametric BvM theorem for BART which previous authors have not done. Next, additional usages of tools from analysis, such as piecewise harmonic functions and piecewise polyhedral multifunctions will probably help to prove the final version of both the BvM theorem for BART and BCF as well as their future extensions to Bayesian causal models. Finally, analysis in our article will prove beneficial in considering future needed extensions and development of area of Bayesian causal inference where a most of the research questions are still nonresponded.

Acknowledgments

Article has been firstly presented in the session Advances in Bayesian causal inference at the 2026 World Meeting of the International Society of Bayesian Analysis (ISBA). Author thanks participants of the session for discussion. This work was supported by the Slovenian Research Agency under grant number N5-0402 (ERC Perspectives).

References

[1] Ismaël Castillo and Richard Nickl. 2014. On the Bernstein–von Mises phenomenon for nonparametric Bayes procedures. *Ann. Statist.* 42 (5) 1941-1969. https://doi.org/10.1214/14-AOS1246.

[2] Hugh A. Chipman. Edward I. George and Robert E. McCulloch. 2010. BART: Bayesian additive regression trees. *Ann. Appl. Stat.,* 4 (1) 266-298, March 2010. https://doi.org/10.1214/09-AOAS285.

[3] P. Richard Hahn. Jared S. Murray and Carlos M. Carvalho, 2020.. Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects (with Discussion). *Bayesian Anal.,* 15 (3) 965-1056, September 2020. https://doi.org/10.1214/19-BA1195.

[4] Seonghyun Jeong and Veronika Rockova, 2023. The Art of BART: Minimax Optimality over Nonhomogeneous Smoothness in High Dimension. *Journal of Machine Learning Research*, 24, 337, 1-65, URL: http://jmlr.org/papers/v24/22-0382.htm.

[5] B. J. K. Kleijn, 2021. Frequentist validity of Bayesian limits. *Ann. Statist.*, 49 (1) 182-202, February 2021. https://doi.org/10.1214/20-AOS1952.

[6] Veronika Ročková. Stéphanie van der Pas, 2020. Posterior concentration for Bayesian regression trees and forests. *Ann. Statist.*, 48 (4) 2108-2131, August 2020. https://doi.org/10.1214/19-AOS1879.

[7] Judith Rousseau, 2010. Rates of convergence for the posterior distributions of mixtures of Betas and adaptive nonparametric estimation of the density. *Ann. Statist.*, 38 (1) 146-180, February 2010. https://doi.org/10.1214/09-AOS703.

[8] A. W. van der Vaart. J. H. van Zanten, 2008. Rates of contraction of posterior distributions based on Gaussian process priors. *Ann. Statist.*, 36 (3) 1435-1463, June 2008. https://doi.org/10.1214/009053607000000613.

Validation of Proper Noun Lexemes in the *Digital Dictionary Database of Slovene* Using Wikidata and Large Language Models

Jaka Čibej
jaka.cibej@ijs.si
Jožef Stefan Institute
University of Ljubljana, Centre for Language Resources and Technologies
University of Ljubljana, Faculty of Arts
Ljubljana, Slovenia

Abstract

The *Digital Dictionary Database of Slovene* [7] (DDDS) is a modern digital lexicographic resource designed as a relational database. It is continuously expanded by extracting candidates from automatically tagged text corpora. The lemmatization and tagging errors inevitably present in these corpora were usually manually identified/corrected by human annotators before being included, but the advent of large language models has provided an additional tool that can help reduce the need for manual labor of lexicographers. We present an experiment on the identification and correction of inadequate proper noun lexemes in DDDS by using existing resources and knowledge bases (such as *Wikidata* and *Wikipedia*) and large language models. The results demonstrate that even zero-shot or contextless prompting can provide useful corrections and significantly reduce the amount of routine work.

Keywords

lexemes, proper nouns, Digital Dictionary Database of Slovene, Wikidata, large language models

1 Introduction

The *Digital Dictionary Database of Slovene* (DDDS) [7] is the largest open-access[1] database with machine-readable information on Slovene words, including information on their inflected forms[2] (e.g., *Slovenija* as the lemma and *Slovenija*, *Slovenije*, *Sloveniji*, ... as its inflected forms).[3] As a modern digital lexicographic resource, it is designed as a PostgreSQL[4] relational database. The data have multiple uses: aside from its obvious benefit as a language resource for human users (e.g., to obtain information on how to inflect/pronounce Slovene words and what senses they have) or as a source of examples for other language resources (such as the *Slovenian Normative Guide 8.0*)[5], the database has also been used to improve lemmatization and morphosyntactic tagging models for Slovene corpora [9][10] or to compile datasets for fine-tuning large language models for linguistic and lexicographic tasks [5][6]. In addition, DDDS can be used as a resource for RAG systems and agentic language models. As a fully machine-readable database, it can be linked to existing digital language resources and knowledge graphs (such as *Wikidata* [11]).

DDDS is continuously expanded with new candidates from frequency lists of lemmas extracted from corpora (such as the *Gigafida Corpus of Written Standard Slovene* [8] or the *Trendi Monitor Corpus of Slovene* [6]). However, such lists contain errors due to several factors. First, the lemmatization and morphosyntactic tagging models for standard Slovene perform well, but are still not entirely accurate (F1≈99.11 for lemmatization [10] and F1≈98.27 for morphosyntactic annotations [9]) and errors frequently occur with lexemes not yet included in DDDS [2], resulting in mislemmatized forms (e.g., *Polzelane* instead of *Polzelan* (demonym for a (male) inhabitant of *Polzela*, a settlement in Slovenia). Second, the texts in corpora contain human-made typographical errors (e.g., **Mancehster*, **Mancester*, **Manchaster*, **Manchetser*, **Manchster* → *Manchester*) and misspelled non-standard forms (such as (**Funlandija* instead of *Fundlandija*; from *Nova Fundlandija* 'Newfoundland').

In previous updates to the DDDS, a linguist (or group thereof) would manually validate the extracted candidates by observing their occurrences in corpora and consulting online resources (such as Wikipedia or other encyclopedic resources), especially when the lexeme is unfamiliar and/or could potentially be lemmatized in different ways (for instance, *Malenov* could be a possessive adjective derived from *Malen*; a legitimate masculine proper noun *Malenov*; a mislemmatized feminine proper noun *Malenova*; and so on).

The latest major update within the *Development of Slovene in a Digital Environment* (RSDO) project [1] expanded the database with approximately 220,000 proper noun lexemes extracted from *Gigafida 2.0* [8] (along with approximately 45,000 candidates from other part-of-speech categories). Due to time and resource constraints, the proper noun lexemes were not manually validated (only filtered to exclude the ones with a frequency below 5 to remove outliers). Their inflected forms were automatically generated with the assumption that the majority of candidates were still correctly lemmatized considering the performance of the lemmatization model. Although other parts-of-speech also

[1] The DDDS is available as an SQL dump (the latest version is published at https://wiki.cjvt.si/books/digital-dictionary-database-of-slovene/page/domain-and-data-model-links) or through a public API documented at *CJVT Wiki*: https://wiki.cjvt.si/books/digital-dictionary-database-of-slovene/chapter/rest-api

[2] Slovene is a morphologically rich Slavic language which features inflected word forms with affixes.

[3] The *Sloleks Morphological Lexicon of Slovene* [3], which until now has been presented as the largest open-access database on Slovene morphological data, has been integrated into DDDS as its morphological basis. It is no longer developed separately and exists only as an export of the morphological part of DDDS.

[4] PostgreSQL: https://www.postgresql.org/

[5] *Pravopis 8.0* (*Slovenian Normative Guide 8.0*): https://pravopis8.fran.si/

[6] For instance, the data from DDDS was used to fine-tune the *GaMS-9B-Instruct-Lex* and *GaMS3-12B-Instruct-Lex* models from *GaMS-9B-Instruct* and *GaMS3-12B-Instruct*, respectively. All are available at HuggingFace.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.18

Table 1: Proper noun lexemes in DDDS by status and grammatical gender (masculine, feminine, neuter).

Gender → Status ↓	M	F	N
Automatic	183,862	38,725	691
Manual	5,525	4,668	159

contain errors, we focus on proper noun lexemes in this paper, as they are currently the largest group with the least previously done manual work (and thus the segment that most likely contains large amounts of errors).

Proper noun lexemes in DDDS contain personal names and surnames (*Deborra*, *Zeljković*), demonyms (*Filipinec*, *Ukrajinka*), names of companies (*Microsoft*, *Agrosoršak*), locations (*Gainesville*, *Catanzaro*), and so on. Any errors in lemmas and their morphosyntactic features result in further errors downstream when inflected forms are generated using *Pregibalnik*,[7] a custom-built tool for Slovene lexicon expansion. For instance, an incorrect lexeme masculine proper noun *Gruzijki* (which should be correctly lemmatized as *Gruzijka* 'female inhabitant of Georgia' and tagged as feminine) is incorrectly inflected as **Gruzijki*, **Gruzijkija*, **Gruzijkiju*, ... (as opposed to *Gruzijka*, *Gruzijke*, *Gruzijki*, ...).

The first step toward correcting these errors is to identify the mislemmatized candidates. We present an experiment on validating proper noun lexemes in DDDS using both existing resources as well as large language models. The main goal is to test whether the method can successfully complement the work of lexicographers and to what extent LLMs can be implemented in future updates of DDDS to ensure clean data.

2 Proper Noun Lexemes in the *Digital Dictionary Database of Slovene*

As of the time of writing this paper (June 2026), the *DDDS* contains a total of 233,630 proper noun lexemes.[8] 10,352 (4.4%) were validated manually (the validation included lemmas, morphosyntactic tags, and their inflected forms), while 223,278 were added entirely automatically from the *Gigafida 2.0 Corpus of Written Standard Slovene* [8]. Table 1 shows the distribution of lexemes by status and grammatical gender.[9]

There is a disproportionately high number of masculine lexemes among the automatically added units, which reflects a frequently occurring error of the lemmatization model: the tendency to interpret inflected forms of out-of-vocabulary units as basic forms, along with a bias to tag them as masculine nouns. Multiple candidates should in fact be converged into a single lexeme (e.g., **Ukrajinkama* (feminine), **Ukrajink* (masculine), **Ukrajinko* (masculine) → *Ukrajinka* (feminine); **Filipince* (masculine), **Filipincem* (masculine), **Filipincev* (masculine) → *Filipinec* (masculine)).

It should be noted that in DDDS, some proper noun lexemes (such as surnames) can be both masculine and feminine because they are inflected differently depending on grammatical gender. In such cases, duplicate lexemes are generated (e.g., *Novak* (masculine) and *Novak* (feminine)).

In the following sections, we perform several cross-comparisons with other available resources to first determine whether the lemmas of automatically added candidates are legitimate.

3 Validation with Existing Resources

First, since the automatically added proper noun lexemes were extracted from Slovene texts from *Gigafida 2.0* [8] and contain large amounts of Slovene names and surnames, we cross-compare them with the lists of names and surnames obtained from the *Statistical Office of the Republic of Slovenia* (SURS)[10]. The SURS lists contain the majority of names and surnames used in Slovenia (in their basic forms). Female and male names are listed separately. This comparison resulted in 2,948 lexemes validated by the list of male names, 2,828 confirmed by the list of female names, and 23,141 validated by the list of surnames.[11]

At this stage, we considered further validating lexemes with 196,798 Slovene Wikipedia pages from *Wikipedia Monthly* [4],[12] a repository that provides monthly, multilingual dumps of Wikipedia, processed and prepared for easy use in NLP projects. However, the resource is problematic because all titles of Wikipedia articles are title-cased regardless of whether they are proper nouns or not (**Jezikoslovje* → *jezikoslovje* 'linguistics', **Stegodon* → *stegodon* 'stegodon', **Slovenščina* → *slovenščina* 'Slovene (language)', **Rehabilitacija* → *rehabilitacija* 'rehabilitation'). This ambiguity also frequently occurs with the CLASSLA-Stanza lemmatization/tagging models: common nouns are lemmatized as title-case and tagged as proper nouns (either because of their sentence-initial position or use in multiword named entities), so direct validation using Wikipedia articles would result in false positives. Validation through Wikipedia pages was thus postponed to a later stage.

The capitalization problem does not occur in Wikidata [11],[13] a free open-access knowledge base with over 122 million data entities that is readable by both machines and humans. In Wikidata, each item is listed in its basic canonical form, which makes it more suitable for lexeme validation. The database can be searched through the *Wikidata Query Service*[14] using SPARQL Query Language[15]. We compiled a SPARQL query to test how many of the 223,278 automatically added lexemes from DDDS are present in Wikidata as an item (each query contained a batch of 50 lexemes; the total number of queries was 4,466). The query returned the Wikidata identifier of the queried item itself, as well as the identifier and label of the hierarchically superior item linked to the queried item through the *instanceOf* property (*P31*). An example for the lexeme *Petelinjek* is shown in Table 2. The lexeme is

[7] *Pregibalnik* is available as open-access code at https://github.com/clarinsi/SloInflector and as an API service at https://orodja.cjvt.si/pregibalnik/docs; the *Form Generator* component, which deals with generating inflected forms, is also available as an API at https://orodja.cjvt.si/pregibalnik/form-generator/docs; depending on part-of-speech, the current version achieves an accuracy between 75-98% (see [1])

[8] Within DDDS, there is an important distinction: *lexemes* are single-word units that form the morphological basis for *lexical units* (which can also consists of multiple words). For instance, the lexical unit *Šempeter pri Gorici* (a town in Slovenia) is linked to three lexemes: *Šempeter* (noun, proper, masculine), *pri* (preposition), and *Gorica* (noun, proper, feminine)); lexemes are carriers of purely morphological information (such as inflected forms), while lexical units are semantic in nature (i.e. they are assigned senses, definitions, or links to other knowledge bases).

[9] In line with the *Multext-East v6 Slovene Morphosyntactic Specifications* (https://nl.ijs.si/ME/V6/msd/html/msd-sl.html), the only lexeme-level morphosyntactic feature for proper nouns is grammatical gender.

[10] SURS - SiStat: https://pxweb.stat.si/SiStat/sl

[11] Note that there is some overlap between the categories and some lexemes were validated by multiple lists, potentially warranting multiple lexemes in DDDS. Likewise, the names present on the lists but not included in DDDS can be added during the next database expansion.

[12] The latest export of *Wikipedia Monthly* at the time of writing this paper was from March 2026.

[13] Wikidata: https://www.wikidata.org/

[14] Wikidata Query Service: https://query.wikidata.org/

[15] SPARQL Query Language: https://www.w3.org/TR/sparql11-query/

Table 2: Results of the SPARQL query used to search Wikidata to validate the lexeme *Petelinjek*.

Item	Identifier	InstanceOf	InstanceOfLabel
Petelinjek	Q2400950	Q486972	human settlement
Petelinjek	Q31528946	Q8502	mountain

Table 3: The 10 most frequent labels of queried Wikidata items.

Label	Frequency
family name	81,826
human	40,074
painting	38,264
human settlement	27,497
album	26,340
film	20,688
unincorporated community in the United States	17,413
male given name	16,239
ship	12,839
single	12,027

present in Wikidata under two items; the first is categorized as a human settlement and the second as a mountain.

A total of 134,316 lexemes were validated through Wikidata (approx. 60% of all automatically added lexemes). However, it should be mentioned that the presence of the lexeme in Wikidata does not necessarily indicate that this is a basis for a relevant lexical unit in DDDS. For instance, the query for the lexeme *Arcobaleno* returns 71 identifiers in Wikidata; all of them refer to specific kindergarten and primary school institutions in Italy. None of these are mentioned in the *Gigafida 2.0* corpus, where the occurrences of *Arcobaleno* refer to different associations in Gorica/Gorizia, Italy. The validated lexemes will be further analyzed based on their *instanceOf* labels to determine which ones are present in more relevant categories that are frequently inflected in Slovene (names, surnames, locations), while the overly specific ones can potentially be removed as the DDDS is not meant to be a comprehensive encyclopedic resource, but primarily a linguistic database. The 10 most frequently occurring labels from the queries are shown in Table 3.

At this point, we checked whether any of the queried Wikidata items had links to Slovene Wikipedia pages: for instance, the Wikidata item for *Benelux* contained the link to the Slovene Wikipedia page for *Beneluks*, confirming *Beneluks* as a legitimate lexeme. A total of 15,901 additional lexemes were validated this way.

4 Validation and Correction with Large Language Models

The remaining 77,083 lexemes not yet found in any of the previous resources were analyzed using large language models. In the prompt design phase, different models were tested on a small sample of 200 lexemes, including *GPT 5.4*, *Gemini 3.5 Flash*, *Claude 4.6*, and *GaMS-27B-Instruct-Nemotron*.[16] A qualitative manual analysis of the results showed that the most useful output was provided by *Gemini 3.5 Flash*, as it contained the most informative context for its decisions on lexeme validation, distinguishing between different problems (misspellings, encoding errors, mislemmatizations, etc.) as shown in Table 4. The following prompt was used (with a list of 500 lexemes per prompt):

Table 4: Example of the output of Gemini 3.5 Flash for the proper noun lexeme correction task (comments truncated for brevity).

ID	Original	Corrected	Comment
442307	Surinamec	Surinamec	Legitimate entity.
465061	Koktalj	koktajl	Misspelled form...
465111	Zlatoličja	Zlatoličje	Mislemmatized...

> Please take on the role of a lexicographer with native knowledge of Slovene. You are helping me validate a list of lexemes for the Digital Dictionary Database of Slovene. I will provide you with a list of lexemes and their IDs. Your task is to determine whether each lexeme refers to a legitimate entity or if it is a mislemmatized form. If the entry is not correctly lemmatized, please provide the correct lemma. Make sure you analyze the entire list. It is crucial that you keep the IDs as they are.
> Reply only with a Markdown table structured in the following manner: |lexeme_id|original_lemma|correct_lemma|additional_comment|
> You can add very short comments (in English!) in the table to explain your corrections. No other explanations outside the table are required.
> This is the list of lexemes: <list>

4.1 Results

For 33,343 lexemes (43% of analyzed lexemes), the model confirmed their lemma and made no changes, whereas the lemmas for the remaining 43,740 lexemes (57%) were modified in different ways: 155 lexemes were only converted from lower-case to title-case (**vodafone* → *Vodafone*, **yamaha* → *Yamaha*); 7,859 lexemes were converted from title-case to lower-case (**Kamnolomstvo* → *kamnolomstvo*, **Spopad* → *spopad*); 35,726 lexemes were modified in more complex ways, usually by correcting lemmatization errors in morphology (**Nabucc* → *Nabucco*, **Pensilvani* → *Pensilvanija*, **Lenoro* → *Lenora*), suggesting preferred variants or alternatives (*Moldavijec* → *Moldavec* '(male) inhabitant of Moldova', *Singapurec* → *Singapurčan* '(male) inhabitant of Singapore') or correcting spelling errors (**Podvolovjek* → *Podvolovljek*).

We compared the corrected lexemes to the set of lexemes that were either manually confirmed or validated by any of the previous resources (SURS, Wikidata, Wikipedia). 21,861 of the corrected lexemes were essentially duplicates of existing validated lexemes: for instance, both **Jurkloštr* and **Jurkloštro* were correctly modified as *Jurklošter*, which is already included in DDDS and was validated by Wikidata and Wikipedia. Redundant candidates will be removed from DDDS. The high number of such candidates indicates that the use of large language models can successfully mitigate lemmatization errors when expanding the database with new lexeme candidates from corpora.

4.2 Evaluation

We performed a preliminary evaluation of the results by randomly sampling 250 lemmas that have been validated by the

[16] https://huggingface.co/cjvt/GaMS-27B-Instruct-Nemotron

Table 5: Confusion matrix for the evaluation sample of Gemini 3.5 Flash.

True → ↓ Predicted	Correct Lemma	Incorrect Lemma	∑
Correct Lemma	**218**	32	250
Incorrect Lemma	31	**219**	250
∑	249	251	-

model and 250 lemmas that have been modified (stratified by the categories mentioned in Section 4.1; so 83 lower-to-titlecase lemmas, 83 title-to-lowercase lemmas, and 84 lemmas with more complex modifications). We manually evaluated the sample to assess model performance. The results are shown in Table 5 (A=BA=P=R=F1≈87.40; a baseline majority classifier achieves an accuracy of 50.20%).[17]

Despite the very robust prompt and lack of any context for individual lemmas, in the majority of examples, the model correctly predicted whether the lemma is correct or whether it requires modification. It should be noted that this task is also difficult for human annotators without context or prior knowledge of the lemma in question. In some cases, lemmas can also be ambiguous: for instance, the lexeme candidate *Delgada* can be legitimate as part of the location *Ponta Delgada* (Portugal), but not legitimate if it is in fact an inflected form of the surname *Delgado*. Such examples are not frequent, however: in the evaluation sample, only 22 (4%) lexemes were ambiguous.

We also evaluated the quality of the model's lemma corrections, and the lemmas were correct in a total of 448 examples (89.60%), and even in the ambiguous cases, the correction usually corresponded to one of the valid options. Any ambiguities were also explained in the model's additional comment: for instance, the lexeme *lakota* was corrected as *lakota* 'hunger', but the comment ("*Legitimate common noun (hunger) or tribe (Lakota).*") also predicted it might refer to the Lakota people. The model seemed to perform well regardless of whether the correction was due to a lemmatization error (**Bjerringbra → Bjerringbro*), diacritic or grapheme simplifications (**Szekesfehervar → Székesfehérvár*), capitalization errors (*Mccormack → McCormack*), truncated forms or encoding artifacts (*drĹžava → država*), or misspelling (**Dormund → Dortmund*).

The manual evaluation of 500 lemmas (by a lexicographer) took approximately 5 hours, and the results show that annotations of a similar quality can be achieved with a single prompt, leaving more time for complex lexicographic tasks.

5 Conclusion

We have demonstrated an approach to validating proper noun lexemes in DDDS using both existing resources as well as large language models. The results of the experiment[18] show that the proposed method complements existing workflows used to update language resources with corpus-based data and can be used to (pre-)validate proper noun candidates, which provides reliable results and reduces the workload of lexicographers. The identified corrections will be implemented in the DDDS, and the validated data can also be used to expand Wikidata with missing items (and, by extension, Slovene Wikipedia).

Although we have so far focused on *Gemini 3.5 Flash* because of its performance, additional experiments with other open-access models are warranted, particularly with the *GaMS* family of Slovene models. Additional prompts with more context (such as examples of use from corpora) will be tested, and the validated data from DDDS can be used to compile fine-tuning datasets for lemmatization correction.

In this experiment, we presented only the first step of correcting lexemes (lemma validation). We will test LLMs (as well as simpler machine-learning approaches such as logistic regression and support-vector classifiers) to complement the other steps in the workflow of expanding DDDS, such as determining basic morphosyntactic features of lexemes (such as gender for noun lexemes and aspect for verbs) and inflected forms – for instance, to determine whether a masculine noun is animate or inanimate; a semantic category which determines the accusative singular inflected form: *Kerssenberg* (surname, masculine, animate) → accusative singular *Kerssenberga*; *Kienhofen* (toponym, masculine, inanimate) → accusative singular *Kienhofen*. We leave these steps for our future work.

Acknowledgements

The research presented in this paper was carried out within the research programme *Language Resources and Technologies for Slovene* (P6-0411), the research project *Large Language Models for Digital Humanities* (LLM4DH; GC-0002), and the *CLARIN.SI Research Infrastructure* (I0-E004), all funded by the Slovenian Research and Innovation Agency (ARIS). The author also thanks Kaja Gantar for helping with the evaluation and the anonymous reviewers for their constructive comments.

References

[1] Jaka Čibej. 2023. Leksikon besednih oblik Sloleks. Poročilo projekta Razvoj slovenščine v digitalnem okolju Aktivnost DS1.3. Development of Slovene in a Digital Environment. (2023). https://www.cjvt.si/rsdo/wp-content/uploads/sites/18/2023/06/RSDO_Kazalnik_Sloleks_v2.pdf.

[2] Jaka Čibej and Tina Munda. 2024. Metoda polavtomatskega popravljanja lem in oblikoskladenjskih oznak na primeru učnega korpusa govorjene slovenščine ROG. (2024). doi:10.5281/zenodo.13936390.

[3] Jaka Čibej et al. 2022. Morphological lexicon Sloleks 3.0. Slovenian language resource repository CLARIN.SI. (2022). http://hdl.handle.net/11356/1745.

[4] Omar Kamali and Omneity Labs. 2025. Wikipedia Monthly. (2025). doi:10.57967/hf/6575.

[5] Timotej Knez and Slavko Žitnik. 2026. Slovene Lexicographic QA Fine-Tuning Corpus SloLexQA 1.0. Slovenian language resource repository CLARIN.SI. (2026). http://hdl.handle.net/11356/2116.

[6] Iztok Kosem, Jaka Čibej, Kaja Dobrovoljc, Tomaž Erjavec, Nikola Ljubešić, Primož Ponikvar, Mihael Šinkec, and Simon Krek. 2026. Monitor corpus of Slovene Trendi 2026-05. Slovenian language resource repository CLARIN.SI. (2026). http://hdl.handle.net/11356/2219.

[7] Iztok Kosem, Simon Krek, and Polona Gantar. 2021. Semantic data should no longer exist in isolation: the Digital Dictionary Database of Slovenian. In *9th EURALEX International Congress "Lexicography for Inclusion"*, 81–83. https://elex.is/wp-content/uploads/2021/09/Semantic-Data-should-no-longer-exist-in-isolation-the-Digital-Dictionary-Database-of-Slovenian_Kosem-Krek-Gantar_EURALEX2020.pdf.

[8] Simon Krek et al. 2019. Corpus of Written Standard Slovene Gigafida 2.0. Slovenian language resource repository CLARIN.SI. (2019). http://hdl.handle.net/11356/1320.

[9] Nikola Ljubešić, Luka Terčon, and Jaka Čibej. 2023. The CLASSLA-Stanza model for morphosyntactic annotation of standard Slovenian 2.0. Slovenian language resource repository CLARIN.SI. (2023). http://hdl.handle.net/11356/1767.

[10] Luka Terčon, Jaka Čibej, and Nikola Ljubešić. 2023. The CLASSLA-Stanza model for lemmatisation of standard Slovenian 2.0. Slovenian language resource repository CLARIN.SI. (2023). http://hdl.handle.net/11356/1768.

[11] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57, 10, (Sept. 2014), 78–85. doi:10.1145/2629489.

[17] A, BA, P, R, and F1 refer to accuracy, balanced accuracy, macro-precision, macro-recall and macro-F1, respectively.

[18] The prompts, SPARQL queries and evaluation data are available on Github: https://github.com/jakacibej/sikdd2026_proper_noun_validation

Comparative Analysis of Feature Selection and Imbalance Handling for Semiconductor Defect Detection on the SECOM Dataset

Sidorela Uku
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
Sidorela.Uku@ijs.si

Klemen Kenda
Jožef Stefan Institute
Ljubljana, Slovenia
Klemen.Kenda@ijs.si

Abstract

Semiconductor manufacturing demands reliable automated defect detection, yet applying machine learning to real-world fabrication data is complicated by severe class imbalance, high dimensionality, and missing values. This paper presents a comparative study of feature selection and imbalance handling strategies for defect prediction on the SECOM benchmark dataset (1,567 samples, 591 features, 6.6% defect rate). Unlike much prior work, we enforce strict data integrity by performing train/test splitting before any label-dependent preprocessing, preventing information leakage that can inflate reported performance. We evaluate six experimental flows spanning three feature selection paradigms: a Pearson filter, the Boruta wrapper, and FASTENER, an evolutionary search applied here to SECOM for the first time, combined with SMOTE, ADASYN, and class-weighting strategies using LightGBM and SVM classifiers. Every pipeline is additionally assessed by five-fold cross-validation with within-fold preprocessing. On the held-out split the strongest pipeline reaches a geometric mean (GM) of 0.745, but five-fold cross-validation shows single-split estimates to be optimistic by approximately 0.08 GM, and the three selectors become statistically indistinguishable once the split is varied. Threshold choice matters more than feature selection, with class-weighting and threshold tuning raising GM from 0.524 to 0.721. These findings highlight the importance of methodological rigor and principled pipeline design for industrial defect detection.

Keywords

Semiconductor Manufacturing, Defect detection, SECOM, Class imbalance, Feature selection, LightGBM

1 Introduction

Semiconductor manufacturing generates high-dimensional sensor data throughout hundreds of sequential process steps. Ensuring low defect rates is critical, as faulty products lead to significant economic losses and resource waste. Machine learning (ML) offers the potential to automate defect detection by learning complex patterns from sensor measurements, replacing costly and time-consuming manual inspection. However, applying ML to real-world fabrication data presents several interrelated challenges: severe class imbalance, high dimensionality, missing values, and noisy or irrelevant features.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.125

The SECOM (SEmiCOnductor Manufacturing) dataset [10] is one of the most popular public benchmarks for studying these challenges. It contains 1,567 samples with 590 sensor features and 1 timestamp column, of which only 104 (6.6%) represent defective products. Approximately 4.5% of values are missing, and 116 features have constant values, offering no discriminative information. This combination of properties makes SECOM representative of the difficulties encountered in industrial production data more broadly, and it has been identified as one of the three most commonly used manufacturing datasets in ML research [3]. Nevertheless, a systematic review of publicly available manufacturing datasets found that well-documented datasets with clearly labeled features remain scarce, and SECOM's anonymized feature names limit process-level interpretability [6].

Despite a growing body of literature applying ML to the SECOM dataset, existing studies differ substantially in their preprocessing pipelines, algorithm choices, resampling strategies, feature selection methods, and evaluation protocols. These inconsistencies make a direct comparison of the reported results difficult and open the question of which methodological choices truly contribute to improved defect detection. This paper presents a comparative analysis of six pipelines using three feature selection paradigms and three imbalance handling strategies. This work contributes to existing literature in four ways. First, we show that threshold tuning accounts for a GM difference larger than feature selection. Second, we quantify the optimism of a single-split evaluation in comparison to five-fold split. Third, we separate two forms of selector stability: the Pearson filter recovers an identical feature set under markedly different missing-value handling, yet retains fewer than half its features across resampled splits, locating the instability in the scarcity of defective samples rather than in preprocessing choices. Lastly, we report the first application of FASTENER to SECOM, which gives comparable result to the other selectors nor matches their stability, retaining only one feature across all five folds.

2 Related Work

Research on defect detection for semiconductor manufacturing has progressed in different areas of ML, including classifier selection, feature engineering, and methods to handle imbalanced data. Table 1 summarises prior ML and the design choices for each case. It can be observed that tree ensembles and SVM dominate as the classifier and SMOTE as the most used resampling technique. However, reported results present a span of ROC AUC from 0.72-0.95 on the same dataset, which reflects also that the difference is not just change of methods but also the evaluation protocols.

Park et al. [12] addresses this in their work, on the importance of building a pipeline that ensures no data leakage. They argue that train/test splitting must be carried out before any

Table 1: Prior ML studies on SECOM. Reported results are not directly comparable: metrics, splits, and preprocessing order differ across studies. CW + thr. denotes class weighting with threshold tuning.

Study	Feature selection	Imbalance handling	Classifier	Best reported
[11]	Boruta, MARS, PCA	SMOTE, undersampling	LR, RF, GBT	Prec. 0.915
[1]	MARS/BGT	Oversampling	SVM or NN	Prec. 0.99, TPR= 0.918
[7]	ANOVA	SMOTE, undersampling	RF/MLP/RUSBT/SVC/GDBT	ROC AUC 0.72
[4]	12-method voting (≥3)	under + oversampling	LR/SVM/DTC/RF/GBC/XGB	ROC AUC 0.95
[5]	None/Zero-variance	None/SMOTE-ENN	LR/RF/SVM/NN/EEC/BRFC	Prec. 0.26
[13]	missingness-aware repr.	CW + thr. selection	LR, RF, LightGBM	PR-AUC 0.221
[12]	Pearson $\|r\| \geq 0.10$	ADASYN/SMOTE or KM/BL/SVM+SMOTE	LR/SVC/KNN/DT/RF/XGB	GM 0.73
This work	Pearson, Boruta, FASTENER	SMOTE, ADASYN, CW + thr.	LightGBM, SVM	GM 0.746 (0.660 CV)

preprocessing steps. Since handling missing values or reducing dimensionality on full dataset leaks test information into training. We adopt their pipeline and reproduce their best performing case as our baseline.

Closest to our work, Park and Baek [13] evaluate Logistic Regression, RF and LightGBM with validation-based threshold selection, using SECOM as a supplementary dataset to compare imputed against missingness-aware feature representations. They report that on SECOM the validation-selected threshold yielded zero recall on the held-out test set, attributing this to operating-point instability when only 21 defective samples are available per split. Our study takes SECOM as the primary dataset, compares feature *selection* paradigms rather than feature representations, and quantifies that instability directly through repeated cross-validation.

Recent work has also begun to reformulate the task itself. Bae et al. [2] replace binary pass/fail classification with autoencoder-based sub-class discovery, using four jointly optimised loss components (separation, clustering, repulsion, balance) to decompose defect labels into mechanism-aligned sub-classes; combined with Prototypical Networks this gave a 47.8% relative improvement over a binary few-shot baseline on SECOM, with further validation on proprietary wafer test data.

3 Methodology

This study investigates optimal combinations of feature selection and imbalance handling methods for defect prediction using the SECOM dataset. Following prior recommendation, we ensure data integrity by performing train/test splitting (70/30, stratified) *before* any label-dependent preprocessing. This prevents information leakage that can artificially inflate performance metrics.

3.1 Preprocessing Pipeline

Before splitting the data, we identified and removed features with zero variance (constant values across all samples) and entirely empty features. This step was label-independent, as variance is computed from feature values alone. A total of 116 features were removed, reducing the feature space from 590 to 474. The data were then split using stratified sampling into training (1,096 samples: 1,023 Pass, 73 Fail) and test (471 samples: 440 Pass, 31 Fail) sets. Features with more than 50% missing values in the training data were removed (24 features), and the same mask was applied to the test set. As an additional check, we used LightGBM feature importance to confirm that none of the dropped features ranked in the top quartile of importance.

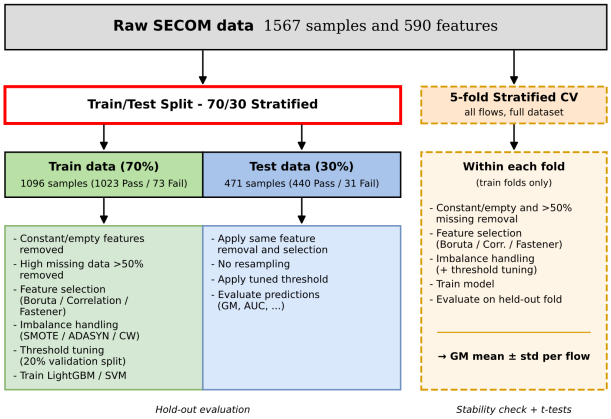


Figure 1: Preprocessing and experimental pipeline overview.

3.2 Feature Selection

We compared three paradigms. *Correlation-based* selection is a filter retaining features with $|r| \geq 0.10$, following Park et al. [12]. *Boruta* is a statistical wrapper using Random Forest (200 trees, max_depth=7, balanced class weights, 100 iterations, α=0.05) that retains features significantly more important than randomized shadow copies. *FASTENER* is an evolutionary wrapper [9] that searches subsets with a mutually guided genetic algorithm, using a base learner of decision-trees (max_depth = 6, balanced class weights) over 100 generations. We ran five seeds and kept features selected at least three times across the five runs. Boruta and correlation each selected 12 features, sharing only 5; FASTENER selected 19.

3.3 Imbalance Handling

We compared three approaches to address the severe 1:14 class imbalance, which differ in whether they modify the training data or not. SMOTE oversamples the minority class by interpolating between nearest neighbors, producing an approximately balanced training set. ADASYN, for comparability with [12], extends SMOTE by concentrating synthetic samples where minority instances are harder to classify. Class weighting build-in mechanism from LightGBM leaves the data unchanged. The scale_pos_weight parameter penalises minority misclassifications in proportion to imbalance ratio. Additionally, the decision

threshold is tuned on a validation split to maximise the geometric mean. All resampling and preprocessing were fitted on training data only, and full parameter settings are provided in code repository in Section 5.

3.4 Model Training and Experimental Design

LightGBM served as the primary classifier due to native NaN handling and efficiency on imbalanced tabular data (n_estimators=500, max_depth=5, learning_rate=0.05, regularization via L1=0.1 and L2=1.0). It uses Gradient-based One-Side Sampling (GOSS), which keeps high-gradient samples and randomly downsamples low-gradient ones to speed up training while preserving accurate split gains. It is also invariant to feature scaling so normalisation is not needed [8].

Six experimental flows were evaluated:

(1) Boruta + SMOTE + LightGBM
(2) Boruta + Class-Weighting + Threshold Tuning + LightGBM
(3) Correlation + MaxAbs + ADASYN + SVM (Park et al. [12])
(4) Correlation + Class-Weighting LightGBM
(5) Boruta + MaxAbs + ADASYN + SVM (isolate feature selection with SVM)
(6) FASTENER + Class-Weighting LightGBM

This design isolates feature selection effects (Flows 2 vs. 4 vs 6; Flows 3 vs. 5) and imbalance handling effects (Flows 1 vs. 2) independently. Since Park et al. [12] already evaluated 150 combinations and identified SVM + ADASYN + MaxAbs as the best-performing configuration, we adopted it directly instead of testing further SVM variants except the combination with Boruta which is new.

3.5 Evaluation

Each model is evaluated using a comprehensive set of metrics: Balanced Accuracy, Recall/Sensitivity, Specificity, Geometric Mean, and AUC. This work focuses on optimizing the Geometric Mean (GM) as a balanced performance of true positive rate (sensitivity) and true negative rate (specificity) as shown in equation 1. Additionally, five-fold stratified cross-validation with within-fold preprocessing validated stability for all flows.

$$\text{GM} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{1}$$

4 Results and Discussion

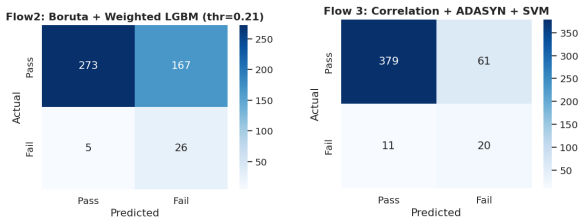
Table 2 summarizes the performance of the test set in all six experimental flows. The results reveal several key patterns regarding the relative importance of imbalance handling and feature selection strategies.

Impact of Imbalance Handling. Flows 1 and 2 share Boruta features and LightGBM but differ in two ways: from resampling method and decision threshold. Flow 2 detects 83.9% defects against 29.0% for Flow 1, and its lower accuracy reflects the 6.6% failure rate rather than a worse model. Additionally, we also tested Flow 2 without threshold tuning, which gave a GM value of 0.636 already better than using SMOTE which is reported in code repository. The GM gap (0.197) is more than four times the AUC gap (0.043) a gap that widens under cross-validation: both pipelines rank samples comparably and diverge mainly in where the boundary is placed. This suggests that the threshold accounts for most of the gain.

Table 2: Test Set Performance Across Experimental Flows. Hold-out versus five-fold cross-validated performance for all six flows, including the standard deviation for GM across folds. Δ the optimism of the single split (hold-out − CV).

	Hold-out				5-fold CV	
Flow	Rec	Spec	GM	AUC	GM mean	Δ
1	0.290	0.946	0.524	0.760	0.408 ± 0.065	+0.116
2	0.839	0.621	0.721	0.803	0.640 ± 0.063	+0.081
3	0.645	0.861	0.745	0.813	0.660 ± 0.088	+0.085
4	0.710	0.648	0.678	0.742	0.633 ± 0.027	+0.045
5	0.613	0.848	0.720	0.804	0.650 ± 0.040	+0.070
6	0.871	0.575	0.707	0.799	0.649 ± 0.042	+0.058

Impact of Feature Selection. Under class-weighted LightGBM (Flows 2, 4, 6) Boruta leads, followed by FASTENER and then Correlation . Under ADASYN and SVM (Flows 3, 5) the order of the first two reverses, with correlation ahead of Boruta. Neither selector is superior independently of the classifier it feeds, and the reversal is comparable in size to the gaps between selectors within either row. Boruta and correlation share only five of twelve features, and FASTENER converges on a third subset of different size. Notably, the correlation filter selected exactly the same 12 sensors as in Park's work, despite our flow following a different approach in handling of missing-values. Yet, under cross validation it retained only five of twelve across all folds. This shows that selection under this filter is robust to preprocessing, but not to training set composition. We additionally tested RFE, PCA and ANOVA within the Flow 1 configuration; all performed worse and are omitted from this report.



(a) Flow 2 (b) Flow 3

Figure 2: Confusion matrix for two of the flows to visualise the different outputs for high number of detected defects with Flow 2 and the highest GM value with Flow 3. Note that the highest recall is from Flow 6, in that case 27 defects were detected but the false positives increase with 20 compared to Flow 2, even though we only detected 1 more.

Moreover, the pipelines differ less in overall quality than in where they sit on the sensitivity vs specificity trade-off. Flow 2 achieves the highest recall (0.839) at the cost of substantially more false positives, while Flow 3 combines moderate recall (0.645) with high specificity (0.861) and correspondingly higher accuracy (Fig. 2). The choice between them depends on the relative cost of false positives and false negatives in the specific manufacturing environment, which also influences the tuning of the classifier towards a selected metric.

Cross-Validation Stability. Table 2 contrasts single-split and five-fold cross-validated performance. Three findings are identified. First, the single split is optimistic for all the pipelines with

less differences in Flow 4 and 6. Which shifts the position on feature selection performance for FASTENER being in the lead when combined with LightGBM with threshold tuning. Second, with the classifier held constant the cross-validated the three feature selectors do not show significant important differences with paired t-tests. Third, the selectors differ markedly in stability: correlation and Boruta retain 5 and 4 features across all five folds, whereas FASTENER retains 1, its subsets varying from 12 to 26 features per fold. Boruta selected 18.0 features on average against 12 on the single split, and the mean tuned threshold was 0.27 against 0.21, reflecting the sensitivity of both selection and calibration to training sample composition. Lastly, while we reported comparative results

Methodological Observations. A key contribution of this work is demonstrating the importance of proper preprocessing order. By performing all label-dependent operations—feature selection, imputation, oversampling, and threshold tuning—strictly within the training partition, we avoid the information leakage that has affected prior SECOM studies. Our results, while lower in absolute terms than some reported in the literature, more accurately reflect real-world generalization performance. False positives remain the dominant practical limitation. At the tuned operating point, Flow 2 flags 167 of 440 passing wafers to recover 26 of 31 defects, roughly six false alarms for every defect caught; Flow 3 trades recall for precision but still raises three false alarms per defect (Fig. 2). This is inherent to the 1:14 imbalance ratio and the small absolute number of defective samples, and suggests that future work should explore cost-sensitive learning frameworks that explicitly model the asymmetric misclassification costs in semiconductor production.

5 Conclusions and Future Work

This study presented a systematic comparison of feature selection and imbalance handling strategies for semiconductor defect detection on the SECOM dataset, with strict accordance to data integrity principles. From our findings, firstly, the choice of imbalance handling method has a substantially greater impact on defect detection performance than the choice of feature selection method. Class-weighting with threshold optimization raised GM from 0.524 to 0.721 under identical feature selection and classifier conditions, although the closely comparable AUC values indicate that the tuned decision threshold, rather than the resampling strategy, accounts for most of this gain. Second, the three selectors capture fundamentally different signal types—Boruta and correlation share only five out of twelve features, and FASTENER converges on a third subset—yet their cross-validated performance did not show a difference. Third, the selectors differ markedly in stability, the Pearson filter and Boruta retain 5 and 4 features across all five folds, whereas the evolutionary search retains none, its subsets varying from 12 to 29 features per fold. Lastly, single-split estimates proved optimistic by approximately 0.045 to 0.116 GM in five of six pipelines, with FASTENER the exception, demonstrating that strict leakage prevention alone is insufficient without repeated evaluation, and that the variance across folds reflects the inherent challenge of small defect sample sizes in industrial datasets.

Future work should address several limitations. The SECOM dataset's anonymized features prevent process-level interpretation of the selected sensors. Exploring ensemble strategies that combine all feature selection approaches may capture complementary signals. Cost-sensitive learning frameworks should be investigated to address the persistently low precision across all pipelines, and deeper integration with recent advances such as sub-class discovery within the defect population [2] may further improve detection in heterogeneous manufacturing environments.

Acknowledgements

This research is supported by EU funding HE MSCA Project Apriori (GA: 101073551) and ARIS Gravitacija (GC-0001).

Data and Code Availability

The source code and experimental configurations are publicly available at https://gitlab.com/Sidorela.Uku/ML_seminar. The SECOM dataset is available from the UCI Machine Learning Repository [10].

References

[1] Ionut Anghel, Tudor Cioara, Dorin Moldovan, Ioan Salomie, and Madalina Maria Tomus. 2018. Prediction of manufacturing processes errors: Gradient boosted trees versus deep neural networks. In *2018 IEEE 16th International conference on embedded and ubiquitous computing (EUC)*. IEEE, 29–36.

[2] Young-Mok Bae, Yingdong He, Zhen He, and Kwang-Jae Kim. 2026. Sub-class discovery in semiconductor defect detection through clustering and few-shot learning. *Expert Systems with Applications* (2026), 131353.

[3] Alican Dogan and Derya Birant. 2021. Machine learning and data mining in manufacturing. *Expert Systems with Applications* 166 (2021), 114060. doi:10.1016/j.eswa.2020.114060

[4] Abdelrahman Farrag, Mohammed-Khalil Ghali, and Yu Jin. 2024. Rare class prediction model for smart industry in semiconductor manufacturing. *arXiv preprint arXiv:2406.04533* (2024).

[5] Daksh Gandhi, Rajat Dandekar, Sreedath Panat, and Raj Dandekar. 2026. Enhancing Failure Detection in Semiconductor Manufacturing Using Balanced Random Forest Model. In *International Conference on Machine Learning and Soft Computing*. Springer, 625–635.

[6] Valentina Golendukhina, Bianca Wiesmayr, and Michael Felderer. 2024. A review of publicly available datasets from manufacturing systems. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–8.

[7] Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. 2022. Product failure detection for production lines using a data-driven model. *Expert Systems with Applications* 202 (2022), 117398.

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[9] Filip Koprivec, Klemen Kenda, and Beno Šircelj. 2020. Fastener feature selection for inference from earth observation data. *Entropy* 22, 11 (2020), 1198.

[10] Michael McCann and Adrian Johnston. 2008. SECOM. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54305.

[11] Dorin Moldovan, Tudor Cioara, Ionut Anghel, and Ioan Salomie. 2017. Machine learning for sensor-based manufacturing processes. In *2017 13th IEEE international conference on intelligent computer communication and processing (ICCP)*. IEEE, 147–154.

[12] Ha-Je Park, Yun-Su Koo, Hee-Yeong Yang, Young-Shin Han, and Choon-Sung Nam. 2024. Study on data preprocessing for machine learning based on semiconductor manufacturing processes. *Sensors* 24, 17 (2024), 5461.

[13] Jung Kyu Park and Youngmi Baek. 2026. Data-Driven Defect Prediction for Manufacturing Quality Monitoring Under Class Imbalance and Missing Data: A Performance–Efficiency Trade-Off Analysis. *Machines* 14, 7 (2026), 716.

Operationalising Active Learning for Power Grid Security Assessment: A Human-in-the-Loop Retraining Service for a Digital Twin Platform

Gašper Leskovec
leskovecg@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Costas Mylonas
kmylonas@ubitech.eu
UBITECH
Athens, Greece

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Active learning (AL) can reduce the number of expensive N-1 contingency simulations needed to train data-driven power grid security classifiers, but the reported gains come almost entirely from offline studies on a static pool of pre-computed operating points, with no human in the loop. This paper reports the integration of an AL-based security assessment service into an operational digital twin platform, describing the deployed architecture, the dashboard–backend interface contract, and the resulting workflow: day-ahead candidates are ranked by informativeness, the operator selects which of them justify simulation, and the labels are produced by the digital twin rather than by the human. The loop has so far been exercised manually — 31 retraining runs and 91 ranking calls over twelve distinct day-ahead sets, with no scheduled operation — so what we report is system behaviour and not label efficiency. Under a temporal split, aggregate discrimination is nearly unchanged while recall on the insecure class falls by ten points at the fixed decision threshold; the three uncertainty-based query strategies coincide in the binary setting, so an interface offering four options realises two behaviours; and the top of the ranking reorders under retraining that leaves the reported model quality unchanged.

Keywords

active learning, human-in-the-loop, digital twin, power grid security assessment, N-1 criterion, system integration

1 Introduction

Transmission grids are operated to withstand the loss of any single component [3]. Verifying this N-1 criterion is computationally demanding, since every candidate operating point requires a base-case power flow and one additional power-flow solution per credible contingency. Data-driven classifiers offer a fast alternative: trained on simulated operating points, they predict whether a state is secure at a fraction of the cost of simulating it [1]. The obstacle is the training data. Every label is the product of the same expensive simulations the classifier is meant to avoid.

Active learning (AL) reduces this cost by letting the model choose which points to label [6]. In the power system studies we are aware of, the reported gains are obtained offline, on a static pool, with a simulator queried automatically and no human between model and oracle. Day-ahead operation changes the problem: candidates arrive as a small daily set rather than a large pool, an operator decides which are worth simulating, and the model is retrained repeatedly over time.

This paper reports on that transition. We contribute a deployed architecture that turns an offline AL study into an operational component of a digital twin platform, comprising candidate ranking, simulator-based labelling and cumulative batch retraining; an interface contract for *operator-as-selector* human-in-the-loop AL, in which the human decides what is worth simulating while the digital twin provides the labels; and a report of what the recorded operation of that loop does and does not support. No label-efficiency experiment in deployment is reported: the loop has not yet accumulated the labels such an experiment requires.

2 Background and Case Study

2.1 Related work

In power systems AL has been applied to stability assessment [7] and, within a digital twin architecture, to day-ahead load forecasting with an operator-set query threshold [4]; we differ from the latter in that the task is security classification and the operator sets the simulation budget rather than a model parameter. Our previous work showed, on a digital twin of the Greek transmission network, that pool-based AL matches the classification quality of random sampling with roughly half the labelled operating points [2], and this paper carries that formulation into deployment.

2.2 Case study

The test system is a steady-state digital twin derived from the topology of the Greek transmission network (35 buses, 46 lines, 135 generators, 110 static generators, 20 loads), solved with AC load flow in `pandapower`. Each operating point is screened against 180 single-component contingencies: all line outages except one, and all generator outages. It is labelled *secure* if the base case and every contingency keep bus voltages within $[0.90, 1.10]$ p.u. and line loading below 100 %. A non-convergent power flow is also labelled insecure, and the label does not distinguish a violated limit from a non-convergent solution.

The reference table covers the 8 760 hours of 2023 and is approximately balanced (51.2 % secure). As delivered it held nine extra rows duplicating hours of 1 January, which we removed; seven were exact copies, but two carried conflicting labels for identical feature vectors. All figures below use the deduplicated table; full details of the offline study are in [2].

The classifier sees only the operating point: the active power set-point of every load, conventional generator and static generator, 265 variables, the same set used in [2]. Under a fixed topology and control configuration an operating point is specified by its injections and demands, so a classifier given those variables addresses the same problem the simulator does, from the same starting information; everything recorded during or after the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.108

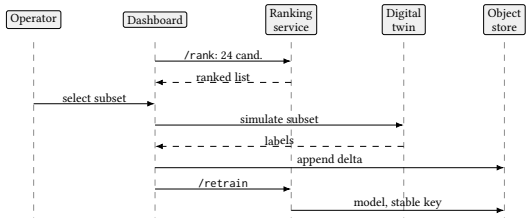


Figure 1: Deployed human-in-the-loop active learning workflow. The operator decides which candidates are simulated; labels come from the digital twin. The ranking service ranks from its own local copy of the current model, so no model is fetched from the store during a call.

simulation is withheld. The two conflicting rows show that the qualification is needed: identical vectors received different verdicts, implying a non-zero error floor. Aggregate outcomes such as worst line loading, against which the label is defined, are withheld; an early version trained on them reached perfect held-out accuracy, a symptom rather than a result.

3 Deployed System Architecture

3.1 Components

The system spans two organisations. The operator dashboard, the orchestration backend and the digital twin simulator are run by the industrial partner; a shared object store is the exchange layer; the ranking and retraining service runs at JSI as a containerised application (Figure 1). Neither side holds a copy of the other's system: they are coupled only through a small HTTP contract and agreed object keys, since the two codebases are released independently.

3.2 Service interface

The service exposes two operational endpoints behind a shared API key held as a deployment secret.

Candidate ranking. `POST /rank` receives the day-ahead candidates, one per forecast hour, each with a timestamp and the full 265-dimensional feature vector, together with the operator's chosen strategy; the dashboard supplies the vectors, and the service does not observe whether they are forecasts or replayed profiles. It ranks them against its local copy of the current model, returning each candidate with an informativeness score and a rank. Four strategies are exposed: least-confident, margin and entropy sampling, and a fourth, named `random` in the interface, which is in fact a deterministic passthrough assigning a constant score and so preserving submission order. Since candidates are submitted in clock order, taking the top k under that option returns the first k hours of the day, which is not a neutral sample. The name is a legacy of the offline study; it must not be read as the randomised baseline of [2], and renaming it is pending.

Retraining. `POST /retrain` retrieves the newly labelled rows from the object store, merges them with the base table held locally, refits a random forest of 400 trees with a fixed seed, and publishes the model with its metrics.

3.3 Engineering decisions

Two decisions were forced by deployment rather than design, and are the parts of this work most likely to transfer.

Delta-based exchange. Transferring the full training table every cycle proved unreliable: the shared object-store gateway timed out on larger payloads. Only new labelled rows are transferred; the base table is held locally and merged at training time, and the published model is likewise cached on local disk and served from there. The service is therefore not stateless, a trade-off between reproducibility and transfer reliability, and nothing pins the contents of the base table; a checksum would close that gap cheaply.

Logging, and what the operator controls. Every call is appended to a structured log with its configuration and outcome — for ranking, candidate timestamps and scores but not feature vectors; log writes are non-fatal. This record is the evidence base for Section 5. What the log does not contain is any operator-set model parameter: the operator controls how many candidates are simulated and nothing else, while thresholds, tree counts and seeds remain configuration owned by the ML engineer.

4 Operator-in-the-Loop Workflow

The role of the human here differs from the standard AL formulation. The operator does not assign labels but acts as a *selector* exercising operational judgement: the operator inspects the ranked day-ahead candidates and decides which justify the cost of a full N-1 analysis, after which the digital twin produces the reference label that is appended to the training data, so the labelling oracle remains a physics-based simulation. Because the human is not the oracle, a poor selection does not inject an incorrect label; it merely spends the budget on a less informative point, so the failure mode is inefficiency rather than contamination, which justifies giving the operator latitude. The selection is meant to sit alongside established day-ahead security assessment practice [3], the score being weighed against expected loading, planned outages and corridor flows; none of that context reaches the service today.

4.1 What an explanation can and cannot show

What the interface can usefully display is constrained by the model. We computed TreeSHAP attributions for the deployed forest over the 2 208-point temporal test set of Section 5. In aggregate, generator injections account for 61 % of the total mean absolute attribution, static generators for 33 % and loads for 6 %. Read per feature rather than per group, these shares mostly track group size: the three groups differ by less than a factor of two per variable. The model does not rest on a small set of dominant injections, and a global importance ranking offers the operator little that knowledge of the network does not supply.

The per-candidate picture is less conclusive than it first appears. The test point closest to the boundary has a predicted insecure probability of 0.500, which for a nearly balanced training set is about where the model's expected output lies, so attributions there sum to almost nothing by construction and their small, mixed-sign magnitudes follow from that choice of point rather than being evidence about the boundary; all it shows is that no single element was worth displaying, consistent with the aggregate result. Attribution is on this evidence better spent on framing: where a candidate sits relative to the rest of the day, and which candidates are tied with it.

Table 1: Randomised and temporal split of the deduplicated reference table ($n = 8\,760$), same seed and code path in a single execution. Brackets give 95 % Wilson intervals; ROC AUC has none, not being a proportion. The temporal split trains on January–September ($n = 6\,552$) and tests on October–December ($n = 2\,208$); the randomised split trains on 7 008 rows and holds out 1 752.

Metric	Randomised split	Temporal split
Accuracy	0.941 [0.929, 0.951]	0.910 [0.897, 0.921]
Recall (insecure)	0.939 [0.921, 0.953]	0.841 [0.818, 0.861]
Precision (insecure)	0.939 [0.921, 0.953]	0.981 [0.971, 0.988]
ROC AUC	0.985	0.978

5 Evaluation and Deployment Observations

5.1 Temporal generalisation

This subsection is an offline analysis of the 2023 reference table, not a measurement of the deployed loop. The offline study of [2] used a time-aware split, but the service reports metrics on a randomised hold-out, which is the usual convention and also the optimistic one: it interleaves training and test across the year, so the model is tested on hours whose neighbours it has seen, whereas a temporal split tests only on the period after a cut date, which is what deployment looks like. We evaluated the same model family under both regimes in one run (Table 1).

The accuracy loss is modest, but the composition of the errors changes: under the temporal split the model misses 181 of 1 135 insecure operating points while raising only 18 false alarms on 1 073 secure ones, whereas under the randomised split the two error types are exactly balanced (52 and 52). Recall falls by about ten percentage points, precision rises by four, and ROC AUC is nearly unchanged. We compare the regimes descriptively, through the intervals in Table 1, rather than with a two-sample test: the randomised hold-out is drawn from the whole year, so it overlaps the October–December test period, and the two proportions come from models fitted on different data — and on different amounts of it, 6 552 rows against 7 008.

The unchanged AUC is consistent with the posterior ordering largely surviving; what degrades is performance at the fixed threshold, where the model becomes systematically reluctant to flag the insecure class and so errs in the operationally unfavourable direction. Two mechanisms can produce this and the data do not separate them: drift in the relation between injections and security, and a shift in the class prior, the training period being 47.9 % insecure against 51.4 % in the test period. Both metrics are reported at the default threshold of 0.5, which the service uses and which we did not tune. Under this single cut the loss is attributable to threshold placement rather than to a degraded ranking, so the cheaper remedy is recalibration on a recent labelled window rather than retraining; that window is not free of labels, but it is far smaller than a retraining set.

The deployed model is fitted on 2023, while the candidates it ranks are 2026 hours without digital twin labels, so its accuracy there cannot be measured. Its own confidence gives a label-free indication: over the 288 candidates of the twelve 2026 sets, the mean posterior of the predicted class is 0.78 and half fall below 0.8, against an accuracy of 0.94 on held-out 2023 hours. A calibrated model with that confidence would be about 0.78 accurate; forests are not calibrated by construction, and the service cannot tell forecasts from replayed profiles, so this indicates rather than measures a gap.

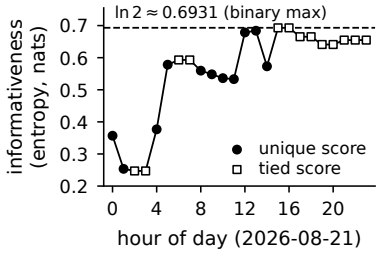


Figure 2: Informativeness scores returned by `/rank` for the twenty-four candidate hours of 2026-08-21, in clock order rather than rank order. Open markers denote tied candidates; the dashed line is the maximum $\ln 2$ for a binary classifier.

5.2 Ranking behaviour on full-day candidate sets

The uncertainty strategies coincide. Least-confident, margin and entropy sampling are distinct in the multi-class case and are offered as distinct choices in the interface, inherited from the option set of the offline study [2]; for binary classification they are not, a reduction already noted in the AL literature [5, 6]. Writing $q = \max(p, 1-p) \in [0.5, 1]$ for the posterior of the predicted class, and negating the margin so all three are maximised,

$$s_{\mathrm{LC}} = 1 - q, \qquad s_{\mathrm{M}} = -(2q - 1), \qquad s_{\mathrm{H}} = H(q), \tag{1}$$

with H the binary entropy in nats. Each is a strictly decreasing function of q alone, so the three induce exactly the same ordering of any candidate set, including the same ties. The offline study observed learning curves lying almost on top of each other, with a small residual gap [2]; it follows from (1) that any such gap originates outside the query policy — in tie resolution, in rounding of the score transform, or in non-determinism in the learner. What deployment adds is the operational consequence: four options realise two behaviours, a single uncertainty ordering and the passthrough. Widening the strategy space requires leaving the uncertainty family altogether.

Scores are well spread but coarsely quantised. One might expect a model fitted on several thousand operating points to assign nearly identical posteriors to all twenty-four hours of a day, leaving nothing to order. It does not. Across the twelve full-day sets the within-day range of scores lies between 0.331 and 0.637 nats, and on seven of them the top candidate comes within 0.001 of $\ln 2 \approx 0.6931$. The scores are, however, coarsely quantised: a day of 24 candidates resolves into only 16 to 21 distinct values, in two to six groups of exact ties, which reach the top three on three of the twelve days. With trees grown to purity the forest averages leaf proportions of 0 or 1 and so resolves posteriors on a grid of $1/400$, and 178 of the 182 distinct entropy values returned lie exactly on it.

Top candidates cluster, but not in a fixed part of the day. Figure 2 plots one call in clock order. Across the twelve full-day sets, the five top-ranked hours lie on average 4.4 h apart in clock distance, against 6.3 h for five hours drawn at random (permutation test, $p < 0.001$). Which hours are favoured, however, varies: pairwise Spearman correlations between the hourly orderings of different days range from −0.84 to +0.87, with a median of +0.27, so the demand-peak shape of the day in Figure 2 does not recur reliably. Informativeness is assessed per candidate within a single day,

with no term rewarding diversity, so a budget spent on top-ranked candidates would draw labels repeatedly from a narrow band of adjacent hours.

The ranking is not stable under retraining. One 24-hour candidate set was ranked repeatedly over a morning with eight retraining runs interleaved, during which the training table grew by 13 rows out of about 8 800. The five top hours at the first and the last call have none in common (Spearman +0.05), while the accuracy and ROC AUC of the successive models varied within one percentage point. With one small unexplained exception, calls with no data-changing retraining between them returned identical scores, so the change comes from the refitted model, although with input vectors unlogged an input change is not strictly excluded. The operator can read all 24 candidates; what the ranking offers is guidance on spending a limited simulation budget, and its top is fragile enough that near-tied candidates are better shown as a group and the ranking regenerated after every refit.

5.3 Scale of the deployed loop

In operation since late June 2026, the service has recorded 31 retraining runs and 91 ranking calls, 89 of them on a full twenty-four-hour candidate set, covering twelve distinct sets from January, August and September 2026. All were issued manually during integration testing: the dashboard exposes the ranking step but does not yet call it on a day-ahead schedule, and the 36 rows appended over the whole deployment were added while exercising the interface rather than through operator-driven simulation. This bounds every claim made above: each step behaves as specified under real inputs, but a few dozen rows on a base of several thousand cannot move the model's aggregate quality, and run-to-run variation in the reported metrics reflects the resampled hold-out rather than anything learned. The frequency of operator interaction, not the ranking policy, is the binding constraint.

6 Limitations and Future Work

What we report is system behaviour under real inputs, so no claim about label efficiency in deployment follows from it. The temporal comparison is a single train/test cut of one year and does not separate drift from the seasonal prior shift; a rolling-origin evaluation would. The 2023 to 2026 gap is bounded in Section 5.1 through the model's confidence but not measured, which would require labels for 2026 hours. The label records only the verdict, not whether an insecure outcome followed from a violated limit or a non-convergent solution, so the classifier is in part fitted to solver behaviour. The explanation results of Section 4 rest on one boundary candidate and on aggregate attributions, not on a study with operators. The forest uses library defaults apart from 400 trees and a fixed seed; the grid model and the reference table are partner-owned and not public.

Within the uncertainty family there is nothing left to compare in the binary case, so committee-based or diversity-aware batch selection is the only direction that changes the ordering, and the clustering of top candidates makes it the more urgent. Exact ties at the top are resolved by submission order; a principled tie-break, or presenting ties as an unordered group, would avoid conveying a precision the score lacks. Threshold recalibration on a recent window and risk-aware selection are the remaining directions, the latter requiring per-sample grid metadata from the twin. Finally, the case study is a single 35-bus system with single-component contingencies; larger topologies, N-k sets and dynamic assessment all raise the cost of a label and strengthen the case for selective simulation.

7 Conclusion

We integrated pool-based active learning for N-1 security assessment into an operational digital twin platform, with the operator selecting the day-ahead points to simulate and the twin producing the labels. The loop has so far been driven manually, so what we report is its behaviour rather than its label efficiency. Under a temporal split, aggregate discrimination is nearly unchanged while recall on the insecure class falls at the fixed threshold, so the randomised hold-out the service reports on is the optimistic case and the threshold is what most needs periodic review. The three uncertainty strategies induce the same ordering in the binary case, so an interface offering four of them offers two behaviours, one of which simply returns the first hours of the day. And the top of the ranking clusters in adjacent hours and reorders under retraining that leaves aggregate quality unchanged, so a fixed budget needs a diversity term and the ranking should be regenerated after every refit. The next steps are threshold recalibration, diversity-aware selection, a principled treatment of ties, and scheduled day-ahead operation, so that label efficiency can be evaluated under operational conditions.

Acknowledgments

This work was supported by the EU-funded HumAIne project [grant number 101120218]. The authors acknowledge the use of LLMs for language editing; all conceptual frameworks, analyses and interpretations remain the sole responsibility of the authors.

References

[1] José-María Hidalgo Arteaga, Fiodar Hancharou, Florian Thams, and Spyros Chatzivasileiadis. 2019. Deep Learning for Power System Security Assessment. In *2019 IEEE Milan PowerTech*. IEEE, 1–6. doi:10.1109/PTC.2019.8810906

[2] Gašper Leskovec, Costas Mylonas, and Klemen Kenda. 2025. Active Learning for Power Grid Security Assessment: Reducing Simulation Cost with Informative Sampling. In *Proceedings of the 28th International Multiconference Information Society – SiKDD 2025*. Ljubljana, Slovenia. doi:10.70314/is.2025.sikdd.11

[3] Kip Morison, Lei Wang, and Prabha Kundur. 2004. Power System Security Assessment. *IEEE Power and Energy Magazine* 2, 5 (2004), 30–39. doi:10.1109/MPAE.2004.1338124

[4] Costas Mylonas, Titos Georgoulakis, and Magda Foti. 2024. Facilitating AI and System Operator Synergy: Active Learning-Enhanced Digital Twin Architecture for Day-Ahead Load Forecasting. In *2024 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 1–6. doi:10.1109/SEST61601.2024.10693980

[5] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to Measure Uncertainty in Uncertainty Sampling for Active Learning. *Machine Learning* 111, 1 (2022), 89–122. doi:10.1007/s10994-021-06003-9

[6] Burr Settles. 2009. *Active Learning Literature Survey*. Technical Report 1648. University of Wisconsin–Madison.

[7] Zhongtuo Shi, Wei Yao, Yong Tang, Xiaomeng Ai, Jinyu Wen, and Shijie Cheng. 2024. Intelligent Power System Stability Assessment and Dominant Instability Mode Identification Using Integrated Active Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems* 35, 7 (2024), 9970–9984. doi:10.1109/TNNLS.2023.3238168

Od kvantnega strojnega učenja do kvantne umetne inteligence: sistematični meta-pregled

From Quantum Machine Learning to Quantum Artificial Intelligence: A Systematic Meta-Review

Aleksandar Brodschneider
Univerza Alma Mater Europaea
Fakulteta ECM
Maribor, Slovenija

Matjaž Gams
Institut »Jožef Stefan«
Jamova 39
Ljubljana, Slovenija

Povzetek

Prispevek obravnava razvoj kvantnega strojnega učenja (QML) kot algoritmične, metodološke in konceptualne podlage za širši razmislek o kvantni umetni inteligenci (QAI) ter njeni možni generativni razširitvi (QGAI). Raziskava je zasnovana kot sistematični meta-pregled drugega reda po načelih PRISMA 2020. Po preverjanju bibliografske in vsebinske neodvisnosti je bilo v primerjalno analizo vključenih 17 preglednih oziroma pregledno-analitičnih virov iz obdobja 2017–2025. Sinteza kaže tri prevladujoče smeri: premik k eksperimentalnemu preverjanju na napravah NISQ, hitro širjenje hibridnih kvantno-klasičnih arhitektur ter razvoj kvantnih generativnih modelov. Rezultati ne potrjujejo QAI kot empirično uveljavljene discipline, temveč kažejo, da QML postopoma ustvarja tehnično in konceptualno osnovo za njen nadaljnji razvoj.

Ključne besede

kvantno strojno učenje, kvantna umetna inteligenca, kvantno-generativna umetna inteligenca, hibridni kvantno-klasični modeli, NISQ, sistematični meta-pregled

Abstract

This paper examines the development of quantum machine learning (QML) as an algorithmic, methodological and conceptual basis for a broader consideration of quantum artificial intelligence (QAI) and its possible generative extension (QGAI). The study applies a second-order systematic meta-review based on PRISMA 2020. After checking bibliographic and substantive independence, 17 review and review-analytical sources published between 2017 and 2025 were included. The synthesis identifies three dominant directions: a shift towards experimental validation on NISQ devices, the rapid growth of hybrid quantum-classical architectures, and the development of quantum generative models. The results do not establish QAI as an empirically mature discipline; rather, they indicate that QML is gradually forming its technical and conceptual foundation.

Keywords

quantum machine learning, quantum artificial intelligence, quantum-generative artificial intelligence, hybrid quantum-classical models, NISQ, systematic meta-review

1 Uvod

Kvantno računalništvo se je iz pretežno teoretičnega področja razvilo v raziskovalno okolje, v katerem se preizkušajo novi algoritmi, programska ogrodja in eksperimentalne aplikacije. Na presečišču kvantne teorije informacij in strojnega učenja se je oblikovalo kvantno strojno učenje (QML), ki preučuje, ali lahko kvantni sistemi ponudijo drugačne načine kodiranja podatkov, prepoznavanja vzorcev, optimizacije in generiranja porazdelitev [1]. Njegov pomen ni omejen na možnost hitrejšega računanja. Kvantni formalizem uvaja reprezentacije, v katerih imajo osrednjo vlogo superpozicija, interferenca, prepletenost in meritev, zato odpira tudi širša vprašanja o naravi informacije in učenja.

Iz tega izhaja razlikovanje med QML in kvantno umetno inteligenco (QAI). QML je danes predvsem operativno-algoritmično področje: vključuje kvantna jedra, variacijske algoritme, parametrizirana kvantna vezja, kvantne nevronske mreže in hibridne postopke. QAI pa je širši konceptualni okvir, ki vprašanje kvantnega računanja povezuje z agentnostjo, prilagajanjem, odločanjem, reprezentacijo znanja in pomena [2, 3]. QAI zato ne nadomešča QML, temveč razširja vprašanje od uporabnosti posameznega algoritma k pogojem, pod katerimi bi kvantno podprt sistem lahko izkazoval kompleksnejše inteligentno delovanje.

Poseben raziskovalni izziv predstavlja razmerje med QML in sodobnimi generativnimi modeli. Veliki jezikovni modeli uspešno modelirajo statistične zakonitosti jezika, vendar njihove zmogljivosti same po sebi niso dokaz semantičnega

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.134

razumevanja. Kvantni generativni modeli, zlasti kvantne generativne nasprotniške mreže (QGAN), pa raziskujejo učenje in vzorčenje kompleksnih verjetnostnih porazdelitev [7, 8]. Možno povezovanje teh smeri je v raziskavi poimenovano kvantno-generativna umetna inteligenca (QGAI), vendar izključno kot razvojna hipoteza, ne kot že vzpostavljena disciplina.

Namen prispevka je povzeti sistematično analizo pregledne literature in odgovoriti na tri vprašanja: v kolikšni meri raziskave QML že vključujejo koncepte QAI; kateri metodološki pristopi prevladujejo pri empiričnem preverjanju QML in kako zanesljivi so; ter ali lahko povezava QML z velikimi jezikovnimi modeli utemelji prihodnji razvoj QGAI. Osrednja teza je previdna: QML je najtrdnejša empirična in algoritmična osnova za razpravo o kvantno podprti inteligenci, vendar sedanji dokazi še ne omogočajo sklepa, da je QAI tehnološko ali empirično uveljavljena.

2 Teoretično izhodišče

Kubit je lahko predstavljen kot superpozicija osnovnih stanj, večkubitni sistemi pa lahko tvorijo prepletena stanja, katerih korelacij ni mogoče učinkovito opisati z enostavnimi klasičnimi modeli. Toda sama superpozicija še ne zagotavlja kvantne prednosti. Uporaben rezultat nastane šele, ko algoritem z nadzorovanimi transformacijami in interferenco okrepi amplitude želenih izidov, meritev pa iz kvantnega stanja pridobi klasično informacijo. Za QML je zato odločilna celotna veriga: priprava in kodiranje podatkov, struktura vezja, optimizacija parametrov, število meritev ter primerjava z ustreznimi klasičnimi modeli.

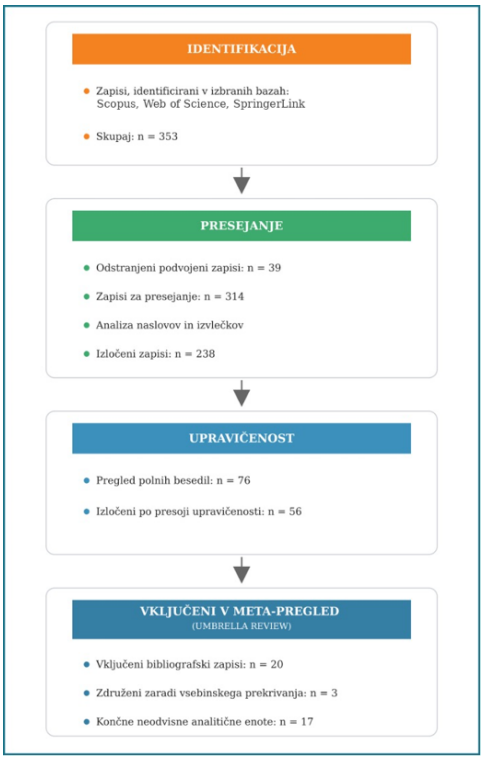
Najpogosteje obravnavani modeli sodobnega QML so variacijski kvantni algoritmi (VQA), kvantne nevronske mreže (QNN), kvantne jedrne metode in hibridne kvantno-klasične arhitekture [4–6]. Variacijski pristop uporablja parametrizirano kvantno vezje za izračun stroškovne funkcije ali pričakovanih vrednosti, klasični optimizator pa na podlagi meritev posodablja parametre. Takšna delitev je primerna za naprave NISQ, ki so omejene s šumom, dekoherenco, številom kubitov, globino vezij in napakami meritev. Hibridnost zato ni zgolj začasni kompromis, temveč praktični računski model, v katerem kvantni procesor prevzema specializirane naloge, klasični del pa nadzor, predobdelavo, optimizacijo in interpretacijo [5].

Prehod od QML k QAI je predvsem konceptualen. V operativnem pomenu se QAI nanaša na kvantno podprte modele učenja, odločanja in prilagajanja. V širšem pomenu pa odpira vprašanje, ali lahko kvantni formalizmi prispevajo k drugačni reprezentaciji konteksta, negotovosti in pomena. Raziskava pri tem ne enači kvantnega procesiranja z zavestjo ali človeškim razumevanjem. QAI obravnava kot teoretični okvir, ki izhaja iz preverljivih dosežkov QML in omogoča znanstveno razpravo o prihodnjih kvantno podprtih agentih in generativnih sistemih.

3 Metodologija

Raziskava je zasnovana kot sistematični meta-pregled drugega reda oziroma »umbrella review«, namenjen kvalitativni primerjalni sintezi pregledne literature, ne statističnemu združevanju primarnih eksperimentov. Postopek identifikacije, presejanja in vključitve virov je sledil izhodiščem PRISMA 2020 [9]. Iskanje je bilo izvedeno 9. oktobra 2025 v bazah Scopus, Web of Science in SpringerLink za obdobje 2013–2025. Osnovni iskalni niz je bil ("quantum machine learning" OR "quantum artificial intelligence") AND ("review" OR "survey" OR "overview") in je bil sintaktično prilagojen posameznim iskalnim vmesnikom, vsebinsko pa je ostal enak. Kjer je bilo mogoče, so bili uporabljeni časovni, področni in dokumentni filtri. Vključeni so bili recenzirani pregledni oziroma pregledno-analitični viri, ki obravnavajo QML, QAI ali hibridne kvantno-klasične pristope, so bili objavljeni v obdobju 2013–2025 in imajo razviden metodološki opis namena, obsega ter načina izbire, razvrščanja ali sinteze literature. Ker v letih 2013–2016 ni bilo najdenih virov, ki bi izpolnjevali vsa merila, končni vzorec zajema obdobje 2017–2025.

Identificiranih je bilo 353 zapisov; po odstranitvi 39 dvojnikov je ostalo 314 zapisov. Po presejanju naslovov in izvlečkov je bilo izločenih 238 zapisov, 76 pa jih je prešlo v pregled polnega besedila. V tej fazi je bilo izločenih 56 prispevkov, ki niso izpolnjevali enega ali več meril glede vrste publikacije, tematske relevantnosti, recenziranosti, časovnega okvira ali razvidnosti metodološkega opisa. Od 20 upravičenih zapisov so bili trije zaradi vsebinskega prekrivanja izločeni, da bi preprečili dvojno štetje in nesorazmerno ponderiranje iste raziskovalne linije; ohranjeni so bili obsežnejši in novejši viri. Končni vzorec je tako obsegal 17 analitično neodvisnih enot.



Slika 1: Postopek izbire virov po načelih PRISMA 2020.

Podatki so bili ekstrahirani v standardizirano Excelovo matriko (avtorji, leto, tema, metodologija, ključni prispevek). Tematsko kodiranje je temeljilo na šestih vnaprej določenih

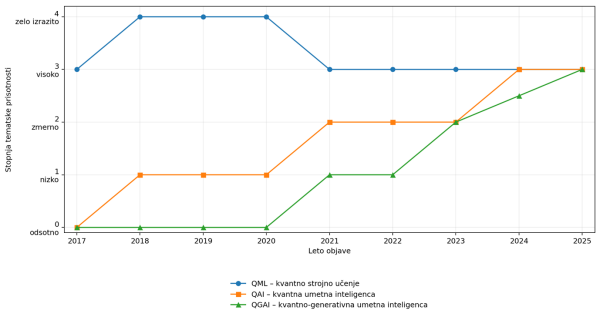
kategorijah: kvantni algoritmi; hibridni kvantno-klasični pristopi; teoretični modeli QAI; aplikativne domene QML; semantični in generativni elementi v povezavi z LLM in QGAI; ter konceptualni prehod QML–QAI–QGAI. Analiza je potekala v treh fazah: tematsko kodiranje, primerjalna sinteza in konceptualno mapiranje. Tematsko kodiranje in primarno analizo je izvedel prvi avtor, kodni in analitični okvir ter interpretacijo rezultatov pa sta v okviru strokovnega pregleda validirala mentor in somentor raziskave. Neodvisno dvojno kodiranje celotnega vzorca in formalni izračun medkoderske zanesljivosti nista bila izvedena. Pri interpretaciji je bila kvalitativno upoštevana tudi metodološka preglednost pregledov: jasnost namena in obsega, preglednost izbire in analize virov, obravnava omejitev ter sledljivost med ugotovitvami in sklepi. To ni predstavljalo formalnega numeričnega ocenjevanja kakovosti. Ker vključeni pregledi sami sintetizirajo širše sklope primarnih raziskav, 17 virov predstavlja analitične enote drugega reda; ugotovljeni trendi zato opisujejo vzorce v analizirani pregledni literaturi in ne statistično reprezentativne ocene celotnega področja QML/QAI.

4 Rezultati

Analiza je pokazala tri prevladujoče razvojne smeri. Prva je premik od teoretičnih in algoritmičnih zasnov k eksperimentalno preverljivim modelom na napravah NISQ. Zgodnja literatura je poudarjala predvsem potencial kvantne linearne algebre, optimizacije, kvantnih podpornih vektorjev in Boltzmannovih strojev [1]. Novejši pregledi pa vse pogosteje obravnavajo parametrizirana vezja, VQA, QNN, kvantne jedrne metode in izvedbe na dejanski ali šumno simulirani strojni opremi [4, 10–14]. Ta premik potrjuje rast eksperimentalne dejavnosti, ne pa še splošne kvantne prednosti.

Druga smer je naraščajoča integracija kvantnih in klasičnih metod. V tipičnem hibridnem sistemu klasična komponenta pripravi podatke, vodi optimizacijo, primerja rezultate in skrbi za interpretacijo, kvantna komponenta pa izvede preslikavo značilk, izračun jedra, variacijsko transformacijo ali generativno vzorčenje. Takšne arhitekture so trenutno najperspektivnejša pot med omejitvami naprav NISQ in prihodnjimi zmogljivejšimi kvantnimi sistemi. Trend se širi tudi v kvantno federirano učenje, kjer se hibridni modeli povezujejo s porazdeljenim učenjem, lokalno obdelavo in varnostnimi mehanizmi [15].

Tretja smer je postopna konceptualna razširitev QML proti QAI in QGAI. V novejših virih so opaznejše teme adaptivnosti, kvantnih učnih agentov, kontekstualnosti in generativnega modeliranja. Najbolj konkretno empirično jedro te razširitve predstavljajo QGAN in drugi kvantni generativni modeli [7, 8]. Vendar tematska prisotnost QAI ali QGAI ne pomeni enake stopnje empirične zrelosti. Graf na sliki 2 je kvalitativna sintezna ocena analiziranih virov, ne bibliometrična meritev ali statistična napoved.



Slika 2: Kvalitativna sinteza tematske prisotnosti QML, QAI in QGAI v izbranih virih.

Odgovor na prvo raziskovalno vprašanje je, da QML že vključuje posamezne elemente, značilne za QAI: adaptivno učenje, kvantne agente, hibridno odločanje, semantično modeliranje in generativne pristope. Ti elementi pa še ne tvorijo enotnega, empirično potrjenega modela QAI. QML je zato mogoče razumeti kot metodološko in tehnično podlago, iz katere se lahko razvija širši koncept kvantne inteligence.

Odgovor na drugo raziskovalno vprašanje izpostavlja VQA, QNN, kvantne jedrne metode in hibridne modele kot najpogostejše pristope k validaciji. Njihova znanstvena zanesljivost je omejena z dekoherenco, šumom, majhnim številom kubitov, nestabilnostjo gradientov, stroški meritev, različnim kodiranjem podatkov ter pomanjkanjem standardiziranih podatkovnih množic in primerjalnih protokolov. Primerjave klasičnih in kvantnih modelov pogosto potrjujejo izvedljivost, ne pa dosledne prednosti kvantnega pristopa [10, 11].

Odgovor na tretje raziskovalno vprašanje je najbolj zadržan. Analizirani korpus ne vsebuje neposredno empirično potrjene integracije QML in velikih jezikovnih modelov. Povezava je za zdaj konceptualna: kvantni podsistem bi lahko izvajal generativno vzorčenje, preslikavo značilk ali optimizacijo, klasični jezikovni podsistem pa obdelavo jezika in interakcijo; integracijski sloj bi povezoval reprezentacije, nadzor in povratne informacije. Tak trislojni okvir je raziskovalna hipoteza, ki zahteva prototipe in eksperimentalno preverjanje.

5 Razprava

Osrednja ugotovitev je, da je treba strogo razlikovati med tehnično izvedljivostjo, merljivo kvantno prednostjo in širšimi trditvami o inteligenci. Uspešna izvedba majhnega QML-modela na kvantnem procesorju dokazuje, da je mogoče določeno arhitekturo implementirati, ne dokazuje pa, da model pri realističnih podatkih prekaša najboljši klasični pristop. Še manj dokazuje agentnost, razumevanje ali zavest. Ta razmejitev je bistvena za preprečevanje prenagljenih sklepov v hitro razvijajočem se področju.

Razvojni lok je mogoče povzeti v treh ravneh. QML je empirična in algoritmična raven, na kateri se razvijajo in primerjajo modeli učenja, optimizacije ter reprezentacije podatkov. QAI je teoretični razširitveni okvir, ki preučuje, kako bi kvantni principi lahko vplivali na agentnost, kontekstualno odločanje in reprezentacijo pomena. QGAI pa je nastajajoča generativna smer, ki povezuje kvantno učenje,

generativno modeliranje in možno semantično reprezentacijo. Ravni so povezane, vendar niso enako zrele: največ empiričnih dokazov ima QML, najmanj pa QGAI.

Raziskava ima več omejitev. Temelji na sekundarnih podatkih in ne vključuje neposrednega preizkušanja algoritmov na realnih kvantnih procesorjih; rezultati so zato odvisni od kakovosti, obsega in metodološke doslednosti obstoječih pregledov. Tematsko kodiranje ni bilo neodvisno podvojeno z drugim koderjem, zato formalna medkoderska zanesljivost ni bila izračunana; interpretativna narava kodiranja, zlasti pri povezavah QML–QAI–QGAI, predstavlja dodatno omejitev. Terminologija QAI, kvantnega agenta, kvantne semantike in QGAI še ni standardizirana, vključevanje velikih jezikovnih modelov pa ostaja konceptualno in ne predstavlja dokaza semantičnega razumevanja.

Prihodnje delo bi moralo dati prednost preverljivim prototipom hibridnih in kvantno-generativnih modelov, transparentnim klasičnim izhodiščem ter ponovljivim eksperimentalnim protokolom. Potrebni so enotnejša terminologija, standardizirane podatkovne množice in metrike, jasnejše poročanje o stroških kodiranja in meritev ter boljša ocena vpliva šuma. Pomembna so tudi vprašanja razložljivosti, nadzora, varnosti, zasebnosti in odgovornosti, zlasti pri porazdeljenih in federiranih kvantno-klasičnih sistemih.

6 Zaključek

Kvalitativna sinteza 17 analitično neodvisnih preglednih virov kaže, da se v analizirani literaturi iz obdobja 2017–2025 QML vse izraziteje usmerja od pretežno teoretičnih zasnov k eksperimentalnim, hibridnim in generativnim pristopom. Hibridne kvantno-klasične arhitekture omogočajo preizkušanje kvantnih komponent v pogojih NISQ, kvantni generativni modeli pa širijo raziskovalni prostor proti QGAI; povezava z velikimi jezikovnimi modeli za zdaj ni empirično potrjena.

Sklep raziskave zato ni, da kvantna umetna inteligenca že obstaja kot samostojna disciplina, temveč da QML oblikuje pogoje za njeno znanstveno obravnavo. QML zagotavlja tehnično in empirično osnovo, QAI konceptualno razširitev, QGAI pa možno generativno nadgradnjo. Nadaljnji napredek bo odvisen ne le od zmogljivejše strojne opreme, temveč tudi od kakovosti eksperimentov, standardizacije, razložljivosti in jasnega razlikovanja med dokazano zmogljivostjo ter konceptualnimi možnostmi.

References / Literatura

[1] Biamonte, J. et al. (2017). Quantum machine learning. Nature, 549(7671), 195–202. DOI: 10.1038/nature23474.

[2] Dunjko, V. and Briegel, H. J. (2018). Machine learning & artificial intelligence in the quantum domain: A review of recent progress. Reports on Progress in Physics, 81(7), 074001. DOI: 10.1088/1361-6633/aab406.

[3] Wichert, A. (2014). Principles of Quantum Artificial Intelligence. Singapore: World Scientific. DOI: 10.1142/8980.

[4] Cerezo, M. et al. (2021). Variational quantum algorithms. Nature Reviews Physics, 3(9), 625–644. DOI: 10.1038/s42254-021-00348-9.

[5] Callison, A. and Chancellor, N. (2022). Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond. Physical Review A, 106(1), 010101. DOI: 10.1103/PhysRevA.106.010101.

[6] Benedetti, M. et al. (2019). Parameterized quantum circuits as machine learning models. Quantum Science and Technology, 4(4), 043001. DOI: 10.1088/2058-9565/ab4eb5.

[7] Lloyd, S. and Weedbrook, C. (2018). Quantum generative adversarial learning. Physical Review Letters, 121(4), 040502. DOI: 10.1103/PhysRevLett.121.040502.

[8] Ngo, T. A., Nguyen, T. and Thang, T. C. (2023). A survey of recent advances in quantum generative adversarial networks. Electronics, 12(4), 856. DOI: 10.3390/electronics12040856.

[9] Page, M. J. et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ, 372, n71. DOI: 10.1136/bmj.n71.

[10] Tychola, K. A., Kalampokas, T. and Papakostas, G. A. (2023). Quantum machine learning—An overview. Electronics, 12(11), 2379. DOI: 10.3390/electronics12112379.

[11] Gujju, Y., Matsuo, A. and Raymond, R. (2024). Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications. Physical Review Applied, 21(6), 067001. DOI: 10.1103/PhysRevApplied.21.067001.

[12] Chen, L. et al. (2024). Design and analysis of quantum machine learning: A survey. Connection Science, 36(1), 2312121. DOI: 10.1080/09540091.2024.2312121.

[13] Peral-García, D., Cruz-Benito, J. and García-Peñalvo, F. J. (2024). Systematic literature review: Quantum machine learning and its applications. Computer Science Review, 51, 100619. DOI: 10.1016/j.cosrev.2024.100619.

[14] Rodríguez-Díaz, F. et al. (2025). A survey of quantum machine learning: Foundations, algorithms, frameworks, data and applications. ACM Computing Surveys, 58(4), Article 91, 1–35. DOI: 10.1145/3764582.

[15] Ballester, R., Cerquides, J. and Artiles, L. (2025). Quantum federated learning: A comprehensive literature review of foundations, challenges, and future directions. Quantum Machine Intelligence, 7, 73. DOI: 10.1007/s42484-025-00292-2.

From Explainability to Accountability: XAI and Fundamental Rights Impact Assessments under the EU AI Act

Lovro Bobnar
Department of International Law
University of Ljubljana, Faculty of Law
Institute for Comparative Law at the Faculty of Law in Ljubljana
Ljubljana, Slovenia
lovro.bobnar@pf.uni-lj.si

Abstract

This paper examines the relationship between Explainable Artificial Intelligence (XAI) and Fundamental Rights Impact Assessment (FRIA) under the European Union Artificial Intelligence Act (AI Act), focusing on whether technical explainability can meaningfully contribute to legal accountability and the protection of fundamental rights. While Article 13 of the AI Act requires sufficient transparency for high-risk artificial intelligence (AI) systems and XAI techniques can support interpretability and human oversight, technical explanations alone cannot establish whether AI-driven decisions are lawful, fair, or non-discriminatory. Article 27 therefore introduces the FRIA as a broader mechanism for identifying and mitigating fundamental rights risks arising from the deployment of high-risk AI systems in specific contexts. However, existing impact assessment approaches may inadequately capture systemic and interconnected harms where rights are assessed in isolation. The paper proposes a three-layer framework connecting technical explanation, operational use and legal justification, informed by intersectionality and vulnerability. A qualitative application to published credit-risk explanations illustrates how XAI findings can inform FRIA enquiries, evidence requests, safeguards and deployment decisions.

Keywords

Explainable Artificial Intelligence; Fundamental Rights Impact Assessment; EU Artificial Intelligence Act; fundamental rights; algorithmic transparency; accountability.

1 Introduction

Regulation (EU) 2024/1689, commonly referred to as the AI Act, establishes a comprehensive framework for the development, placing on the market and use of AI systems in the European Union (EU). Its objectives extend beyond technological safety. The Regulation also seeks to ensure a high level of protection of health, safety and fundamental rights, while supporting innovation and the functioning of the internal market [1].

The AI Act follows a risk-based approach. It prohibits specifically defined practices that conflict with fundamental EU values, including human dignity, freedom, equality, democracy and the rule of law. Examples include certain forms of social scoring and harmful exploitation of vulnerabilities related to age, disability or a person's social or economic situation. High-risk systems are instead subject to the requirements laid down in Articles 8 to 15, including risk management under Article 9, data governance under Article 10, technical documentation and record-keeping under Articles 11 and 12, transparency under Article 13, human oversight under Article 14, and accuracy, robustness and cybersecurity under Article 15. Under Articles 43, 47 and 48, the provider must complete the applicable conformity assessment, draw up an EU declaration of conformity and affix the CE marking as a visible declaration of compliance. The marking does not, however, guarantee that every deployment or AI-assisted decision will comply with fundamental rights. A technically compliant system may still produce unacceptable consequences in a particular institutional setting or when used on a population that differs from the one on which it was tested.

This distinction explains the relationship between Articles 13 and 27 of the AI Act. Article 13(1) requires high-risk AI systems to be sufficiently transparent to enable deployers to interpret their outputs and use them appropriately. Article 27(1), in turn, requires specified deployers to assess the effects that the use of a high-risk AI system may have on fundamental rights. The connection between these obligations is made explicit in Article 27(1)(d), which requires the deployer to identify specific risks of harm while considering the information supplied by the provider under Article 13. XAI may therefore support a FRIA by making the functioning or outputs of a system more intelligible. Nevertheless, explainability and accountability are not equivalent. XAI primarily addresses a descriptive question: how did the system produce a particular output? A legal assessment of the deployment must ask broader questions: was the use of that output lawful and justified, who is responsible for the resulting decision, and what remedies are available if the decision is unlawful or harmful? This paper argues that XAI should serve as a source of technical evidence within a FRIA. It combines legal analysis with a structured qualitative application of a three-layer framework comprising technical explanation, operational explanation and legal justification. Drawing on published credit-

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
http://doi.org/DOI_RECEIVED_AFTER_REVIEW

risk explanations, the application shows how an XAI finding can be connected to its evidential limitations, potentially affected rights, further information requirements and a deployment decision. Published technical findings are distinguished from hypothetical deployment conditions and the author's normative assessment.

2 The Limits of Technical Explainability

Article 13(2) requires the instructions for use accompanying a high-risk AI system to contain concise, complete, correct and clear information that is relevant, accessible and comprehensible to deployers. Article 13(3) further requires information about the system's intended purpose, capabilities, limitations, expected accuracy, foreseeable risks and performance concerning particular persons or groups. Where applicable, the instructions must also include information relevant to explaining and interpreting the system's outputs under Article 13(3)(b)(iv) and (vii), as well as the human-oversight measures referred to in Article 13(3)(d). The provision does not prescribe a particular XAI method or establish a universal metric for explainability. Indeed, the AI Act neither generally mandates the use of XAI nor prohibits black-box models. XAI is therefore one possible means of supporting transparency and human oversight rather than a legally prescribed solution [2], [3].

The appropriate explanation depends on its audience and purpose. Developers may need a global account of the variables and patterns on which a model relies. A doctor, civil servant or credit officer may require an explanation of an individual recommendation and its uncertainty. An affected person may instead need to know which information was decisive, how the output influenced the final decision and how that decision can be challenged. Research therefore supports an audience-specific and lifecycle-based approach [4], [5].

Feature-attribution methods describe the contribution assigned to input features in explaining a model's prediction. SHAP (SHapley Additive exPlanations), for example, indicates how individual input features contribute to a particular prediction relative to a reference prediction [14]. An explanation concerning one individual should be distinguished from a summary of explanations across many individuals: the latter may reveal broader patterns but does not explain every individual decision [15]. For a FRIA, these outputs can help identify features requiring further scrutiny. They do not, by themselves, establish whether reliance on those features is legally permissible or whether the resulting treatment constitutes discrimination [7].

Explanations produced after a model has generated an output, commonly described as *post-hoc* explanations, may approximate its behaviour rather than reveal its actual logic. Their results may vary depending on the method, data or parameters selected. In high-stakes contexts, an inherently interpretable model may therefore be preferable to adding an explanation layer to an unnecessarily opaque system [6]. Explanations should at least be evaluated for fidelity, stability, comprehensibility and their capacity to communicate uncertainty.

Explainability does not independently guarantee non-discrimination. Removing protected characteristics from a dataset does not prevent other variables from functioning as proxies, while historical inequality may already be encoded in labels, institutional practices and definitions of success. Statistical fairness metrics can help identify disparities, but they cannot determine compliance with EU non-discrimination law without contextual legal analysis [7]. More broadly, treating fairness as a purely technical property can obscure the institutional and social environment in which an AI system operates [8].

XAI may even increase risk when a persuasive explanation creates unjustified confidence in an inaccurate output. Article 14(4)(b) expressly requires persons responsible for human oversight to remain aware of the risk of automatically relying or over-relying on an AI system's output. Articles 14(4)(c) and (d) further require them to be able to interpret the output and to decide not to use it or to disregard, override or reverse it. Article 26(2) complements these requirements by obliging deployers to assign human oversight to persons with the necessary competence, training, authority and support. Effective oversight therefore requires more than the nominal presence of a human in the decision-making process [2].

3 The Fundamental Rights Impact Assessment as an Accountability Mechanism

Article 27 shifts attention from the AI system in isolation to its specific context of use. Under Article 27(1), the obligation to conduct a FRIA applies to deployers that are bodies governed by public law, private entities providing public services, and deployers of high-risk systems referred to in points 5(b) and (c) of Annex III. High-risk systems used in the area listed in point 2 of Annex III are excluded from this obligation. Article 27(2) clarifies that the assessment is required for the first use of the system. Following the 2026 amendment to Article 113, third paragraph, point (c)(i), these provisions will apply from 2 December 2027 to systems classified as high-risk under Article 6(2) and Annex III [9]. Private employers are therefore generally outside the scope of Article 27 unless they fall within one of the specifically listed categories.

Article 27(1)(a) to (f) directly defines the required content of a FRIA. Points (a) and (b) require a description of the deployer's processes and the intended duration and frequency of use. Points (c) and (d) require the identification of affected persons and groups and the specific risks of harm, considering the information supplied by the provider under Article 13. Points (e) and (f) require a description of human-oversight measures and the measures to be taken if risks materialise, including internal governance and complaint mechanisms. Under Article 27(2), the assessment must be updated where any of these elements changes or is no longer current.

Article 27(4) permits relevant parts of a DPIA to be incorporated into, or cross-referenced in, the FRIA where they already satisfy the corresponding obligations [9]. The distinction between the assessments should not be reduced to privacy versus other fundamental rights. DPIAs address risks arising from personal-data processing and may also concern rights such as non-discrimination [16]. A FRIA addresses the deployment of the AI system within the scope of Article 27. Existing DPIA findings can therefore be reused, while any outstanding requirements concerning affected groups, deployment-specific harms, human oversight and complaint arrangements must still be addressed.

The principal value of a FRIA is its recognition that rights are affected by a complete socio-technical process. The same score

may create different risks depending on whether it is advisory, presumptive or practically decisive. Its impact also depends on input data, operator incentives, available alternatives and opportunities to challenge the result [8], [10].

Article 27 nevertheless has structural limitations. Its restricted scope may exclude systems that influence important decisions. Providers may understand the model but lack knowledge of its context, while deployers may understand their processes but lack access to technical evidence. The provision also lists assessment elements without establishing a complete methodology or generally requiring the participation of affected persons [10], [11].

A conventional matrix based only on the probability and severity of harm is therefore insufficient. Risks involving absolute rights cannot be accepted merely because a deployment offers economic or administrative benefits. Where limitations of other rights are possible, Article 52(1) of the EU Charter requires respect for the essence of the right, a legitimate objective, necessity and proportionality [12]. The FRIA should consequently combine technical risk analysis with substantive legal assessment. Article 85 complements the preventive function of the FRIA with an *ex post* accountability mechanism by allowing any natural or legal person having grounds to consider that the Regulation has been infringed to lodge a complaint with the relevant market surveillance authority. In addition, under the conditions set out in Article 86(1), affected persons may obtain a clear and meaningful explanation of the role of an Annex III high-risk AI system in a decision producing legal or similarly significant effects and of the main elements of that decision.

4 Intersectionality and Vulnerability

Article 27(1)(c) expressly requires the identification of "the categories of natural persons and groups likely to be affected" in the specific context of use. The provision therefore supplies a legal basis for examining differentiated impacts, but it does not specify how deployers should account for intersecting forms of disadvantage. Assessments organised around isolated categories such as sex, race, age or disability may overlook harms experienced at the intersection of several characteristics. The AI Act also recognises vulnerability in Article 5(1)(b), which prohibits certain AI practices that exploit vulnerabilities related to age, disability or a specific social or economic situation. However, this provision addresses vulnerability only within the narrowly defined context of prohibited exploitative practices and does not provide a general methodology for identifying vulnerability within a FRIA.

A recruitment system might produce similar aggregate outcomes for women and men while disproportionately disadvantaging older migrant women. An education system may achieve high average accuracy but perform poorly for children who combine a linguistic minority background with a learning disability. Small intersectional groups can disappear within aggregate statistics even where the consequences for their members are serious.

Recent scholarship therefore proposes intersectionality and vulnerability as correctives to FRIA models organised around isolated rights and fixed categories [11]. Vulnerability should not be understood only as a permanent personal characteristic. It may also arise from institutional dependency, limited digital literacy, language barriers, economic disadvantage or the absence of realistic alternatives and remedies. The relevant inquiry is not merely which group a person belongs to, but how a particular deployment distributes decision-making power, exposure to harm and the capacity to challenge decisions.

Intersectional analysis should combine disaggregated quantitative testing with qualitative evidence and, where appropriate, participation by affected persons. The Council of Europe's HUDERIA methodology provides a useful model by combining context-based risk analysis, stakeholder engagement, substantive assessment, mitigation planning and iterative review [13]. Although HUDERIA is non-legally binding and cannot replace Article 27, it can help operationalise its relatively open requirements.

5 From Explainability to Accountability

The gap between XAI and accountability cannot be resolved through a single explanation technique. This paper proposes that an effective FRIA should integrate three complementary layers, each anchored in different provisions of the AI Act. Technical explanation, informed particularly by Articles 13 and 15, concerns how the system operates, which variables influence its outputs, how accurate it is, where uncertainty exists and whether performance differs across groups. Operational explanation, reflected in Articles 14 and 26, concerns how the system is integrated into the deployer’s processes, what weight its output carries, who reviews it and when it may be overridden. Legal justification builds on Articles 27, 85 and 86 but extends beyond the AI Act’s minimum requirements by examining the legal basis and objective of the deployment, the rights affected, less restrictive alternatives, the allocation of responsibility and the availability of complaints, explanations and other remedies [5], [10].

To apply these layers, the deployer should first define the intended use and consider whether the proposed system is an appropriate means of achieving it. Each relevant XAI finding should then be documented together with its scope, evidential limitations, potential rights implications, further information requirements, responsible actors and consequences for deployment. This proposed workflow makes the connection between explanation and institutional action explicit.

A published credit-risk study provides a concrete starting point. Bone-Winkel and Reichenbach used SHAP to interpret models developed using lending data from Bondora. Their analysis identified the combined country-and-language feature as the most influential demographic feature and reported different contributions for Estonian-speaking and Russian-speaking Estonian borrowers [17, Section 4.2, Figure 6]. This is a finding about the models examined in that study; it does not establish unlawful discrimination or describe the platform’s actual lending decisions.

For the present illustration, suppose that an EU lender proposes to use a high-risk system exhibiting this explanatory pattern to assess natural persons’ creditworthiness, within Annex III, point 5(b), and Article 27. Assume that an adverse assessment normally leads to refusal unless a loan officer intervenes. The

deployment procedure and the FRIA response below are hypothetical.

At the technical layer, the finding identifies a feature requiring scrutiny. Because country and language are combined, the explanation does not by itself isolate their separate contributions. Moreover, a pattern across explanations cannot establish the reason for a particular refusal. The deployer should obtain documentation explaining the feature's construction, the scope of the explanations and the evidence supporting their reliability in the intended context [15]. The published study supplies a relevant example of explanatory information, rather than validation of the proposed deployment.

At the operational layer, the assessment should examine how the adverse score becomes a lending decision. Relevant questions include whether the loan officer understands the explanation, can correct inaccurate information, has authority to depart from the recommendation and records reasons for doing so. The proposed procedure should also enable applicants to obtain understandable reasons and challenge the decision. Human involvement should therefore be assessed through the reviewer's actual powers and working conditions.

At the legal layer, the explanation raises questions about potentially unequal access to credit. The assessment should identify the applicable non-discrimination rules, examine any connection with protected characteristics and consider any legally available justification and less discriminatory alternatives [7]. It should also investigate whether language-related disadvantage interacts with other circumstances, such as age or economic vulnerability [11]. These are questions for further assessment, rather than conclusions that follow automatically from feature importance.

Assume, finally, that the proposed deployer lacks evidence about how the identified pattern affects lending outcomes for the relevant groups. Under the framework proposed here, this material gap should lead to a recorded decision to defer deployment pending further assessment. Table 1 illustrates how the explanation is translated into a FRIA record.

Table 1. Illustrative FRIA record: the author's proposed application of the published finding in [17].

FRIA element	**Illustrative record**
Affected persons and potential harm — Article 27(1)(c)–(d)	Applicants belonging to the language groups represented by the feature; possible unjustified disadvantage in access to credit.
Evidential limitation	The explanation concerns model behaviour. Evidence connecting the pattern to lending outcomes and its legal significance is still required.
Oversight and complaints — Article 27(1)(e)–(f)	Specify the reviewer's authority, procedures for correcting information, recorded reasons and a route for contesting adverse decisions.
Action and responsibility	The provider supplies technical documentation; the lender's assessment team examines deployment effects and legal implications. The lender retains responsibility for the deployment decision.
Proposed decision	Defer deployment until the material evidence gap has been addressed and the resulting safeguards and residual risks have been assessed.

The three-layer framework complements existing assessment approaches. HUDERIA already connects technical characteristics with deployment context, stakeholder engagement and human-rights impacts [13]. The contribution proposed here is a focused procedure for following an XAI finding through evidential scrutiny, operational use and a reasoned FRIA decision. Relevant DPIA findings can inform this procedure, subject to the requirements discussed in Section 3. The framework therefore provides an analytical structure that can be used within these broader assessment processes.

Affected persons, representative organisations and relevant experts should inform the assessment where appropriate. The proposed documentation and deployment decision process constitutes a good-practice model extending beyond the minimum wording of Article 27. The assessment must subsequently be updated when the elements identified in Article 27(1) change or are no longer current.

6 Conclusion

XAI is an important instrument for transparency, auditing and human oversight, but it cannot independently guarantee compliance with fundamental rights. Technical explanations describe the operation or outputs of an AI system; they do not determine whether its use is lawful, necessary, proportionate or non-discriminatory.

A meaningful FRIA should therefore integrate technical evidence with contextual legal analysis, intersectional identification of affected persons, stakeholder participation and effective remedies. Explainability asks how an AI system produced an output. Accountability asks whether the system should have been used, whether reliance on its output was justified, who is responsible for the decision and what happens when the process causes harm.

The worked example illustrates a procedure for translating published XAI findings into documented FRIA questions and institutional responsibilities. Its contribution is analytical and procedural; whether this approach improves assessments in practice remains a question for empirical evaluation.

References

[1] European Parliament and Council, "Regulation (EU) 2024/1689 laying down harmonized rules on artificial intelligence (Artificial Intelligence Act)," *Official Journal of the European Union*, L 2024/1689, 12 July 2024.

[2] C. Panigutti et al., "The Role of Explainable AI in the Context of the AI Act," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23)*, 2023, pp. 1139–1150.

[3] F. Walke, L. Bennek and T. J. Winkler, "Artificial Intelligence Explainability Requirements of the AI Act and Metrics for Measuring Compliance," in *Solutions and Technologies for Responsible Digitalization*, D. Beverungen, C. Lehrer and M. Trier, Eds., vol. 75. Cham: Springer, 2025, pp. 113–129.

[4] T. Miller, "Explanation in Artificial Intelligence: Insights from the Social Sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[5] E. H. van Beem, "Explainable throughout the AI Lifecycle: The Evolution of the Explainability Provisions in the EU AI Act," *International Review of Law, Computers & Technology*, pp. 1–35, 2026.

[6] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[7] S. Wachter, B. Mittelstadt and C. Russell, "Why Fairness Cannot Be Automated: Bridging the Gap between EU Non-discrimination Law and AI," *Computer Law & Security Review*, vol. 41, Art. 105567, 2021.

[8] A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian and J. Vertesi, "Fairness and Abstraction in Sociotechnical Systems," in *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT '19)*, 2019, pp. 59–68.

[9] European Parliament and Council, "Regulation (EU) 2026/1744 amending Regulation (EU) 2024/1689 as regards the simplification of the implementation of harmonised rules on artificial intelligence," *Official Journal of the European Union*, L 2026/1744, 24 July 2026.

[10] A. Mantelero, "The Fundamental Rights Impact Assessment (FRIA) in the AI Act: Roots, Legal Obligations and Key Elements for a Model Template," *Computer Law & Security Review*, vol. 54, Art. 106020, 2024.

[11] M. Lasek-Markey and L. Hogan, "Delivering on the Promise of the Fundamental Rights Impact Assessments in the EU AI Act: Intersectionality and Vulnerability," *European Journal of Law and Technology*, vol. 16, no. 2, pp. 1–23, 2025.

[12] European Union, "Charter of Fundamental Rights of the European Union," *Official Journal of the European Union*, C 202, pp. 389–405, 7 June 2016.

[13] Council of Europe, Committee on Artificial Intelligence, "Methodology for the Risk and Impact Assessment of Artificial Intelligence Systems from the Point of View of Human Rights, Democracy and the Rule of Law (HUDERIA Methodology)," CAI(2024)16rev2, Strasbourg, 28 November 2024.

[14] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] K. Sokol and P. Flach, "Explainability Fact Sheets: A Framework for Systematic Assessment of Explainable Approaches," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 56–67.

[16] Article 29 Data Protection Working Party, "Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is 'likely to result in a high risk' for the purposes of Regulation 2016/679," WP 248 rev.01, revised and adopted 4 October 2017.

[17] G. F. Bone-Winkel and F. Reichenbach, "Improving credit risk assessment in P2P lending with explainable machine learning survival analysis," Digital Finance, vol. 6, pp. 501–542, 2024.

Influencing Sentiment in LLM Output via Token Selection Manipulation

Kaja Praprotnik*
kaja.praprotnik@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Niki Hrovatin*
niki.hrovatin@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Jernej Vičič*
jernej.vicic@upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Domen Vake*
domen.vake@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Abstract

In this work, we explore whether a static lexicon-derived token bias can steer a Large Language Model toward more positive or negative outputs, without any additional training. On Llama-3.1-8B-Instruct, a lexicon-derived prefix-count ratio lookup table manipulates the logits at each decoding step. Two login manipulations are tested: additive bias and multiplication/division boosting, each at several weights. The additive approach steers sentiment in the positive direction and is scored with the sentiment tool VADER, where the additive bias produces no significant steering. Based on that result, we approach the problem with a more aggressive multiplication/division method. The multiplication/division boosting produces significant sentiment steering in both directions, with small effect sizes. Steering is more effective in the positive direction.

Keywords

Sentiment Analysis, Large Language Models, Sentiment Steering

1 Introduction

Large Language Models (LLM) are incorporated in various fields, including service-oriented industries such as chatbots. Chatbots can be used in healthcare [3] and education [16] as an accessible communication tool. They can also be used as customer support [22] or sales assistant [5], helping users with various tasks. Interacting with a user depends on setting an appropriate tone of the conversation. A positive and uplifting tone could be used in an e-commerce setting, where friendliness and empathy are positively associated with trust in the chatbot [6]. In general, emotionally sensitive chatbots might be perceived as more capable and trustworthy to users in comparison to emotionally insensitive ones and using positive words has a positive impact on users [4, 25]. Sentiment steering also raises ethical concerns, particularly for vulnerable users, as increased trust in emotionally responsive chatbots may heighten susceptibility to influence and encourage overreliance. Responsible use should therefore include transparency about the system's artificial nature, user control over emotional adaptation, and safeguards against manipulative language or exploitation of emotional vulnerability. A detailed examination of these ethical issues is beyond the scope of this paper, but they remain important deployment considerations.

It might be worth exploring different ways of steering LLM outcomes into the right emotional direction. In this work, we explore whether we can steer the sentiment of LLM output by logit boosting based on a sentiment lexicon at each decoding step without any additional model training.

2 Related Work

A few papers are exploring different techniques for approaching the sentiment steering problem in LLMs. By using Controllable Text Generation (CTG) techniques, we can direct the model to reach specific objectives [17]. For example, using reinforcement learning (RL) fine-tuning through human feedback, the work of Ziegler et al. [27] solves the task of using high sentiment or physically descriptive language in stylistic continuation. Another way to simply influence the sentiment of the LLM's output is by prompt engineering. Monafal et al. [20] find that the model can be steered through task-specific instructions (prompts) towards generating summaries with a positive tone. An alternative strategy for controlled language generation is the Plug and Play Language Model (PPLM)of Dathathri et al. [8]. This method combines a pretrained large transformer-based model with simple attribute classifiers. PPLM can be used for language detoxification. A different approach to CTG is activation engineering [23], which uses the Activation Addition (ActAdd) technique. Based on this and on Concept Activation Vectors (CAVs) [15], the paper by Zhang et al. [26] introduce GCAV Framework. GCAV framework trains a CAV for a specific concept, such as toxicity, and then uses it to influence a model to control that same concept. Hiranandani et al. [9] propose Plugin, a token-level probability reweighting framework.

Liu et al. [18] modify logits directly at the decoding time by combining the base model's logits with the logits of additional expert and anti-expert models at each decoding step. The work of Yang and Klein [24] propose a related approach, where the base model's logits are modified at decoding time by combining them with the output of a trained binary classifier.

Close to our approach, An et al. [2] propose SWAI, a training-free steering method that biases token logits, using z-normalized one-vs-rest log-odds scores from labeled corpora. Similar to SWAI, our first approach applies a constant bias value. Unlike SWAI, which restricts biasing to a top-K set and recomputes the bias token set in each step, we do not restrict to top-K and instead select tokens from a precomputed, lexicon-driven token set. We test it on RLHF-instruction-tuned Llama-3.1-8B-Instruct, while

*These authors contributed equally.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.116

SWAI tests it on non-instruction-tuned models. An et al. [1] apply a similar method to apply it to a diffusion instruction-tuned model, where they find a clear asymmetry between toxic and non-toxic steering and suggest that instruction-tuned DLMs have a strong internal resistance to toxic continuations. In our work, we explore whether this asymmetry extends to sentiment steering in an autoregressive instruction-tuned model.

3 Problem formulation

Let T be a set of all the tokens in the tokenizer. Let $W = P \cup N$ be all of the words in the sentiment lexicon [11], where P is a set of all the positive words (e.g. *amusing*, *awesome*) and N is a set of all the negative words (e.g. *agonize*, *brutality*).

We say that a token $t \sqsubseteq_{\text{pre}} w$ for $w \in W$ if t is a *prefix* of a word w for any $t \in T$. Then, let *positive count* ($C^P(t)$) of a token t for any $t \in T$ be the number of the positive words $p \in P$ such that $t \sqsubseteq_{\text{pre}} p$, that is,

$$C^P(t) = |\{p \mid t \sqsubseteq_{\text{pre}} p, p \in P\}|. \quad (1)$$

Similarly, a *negative count* $C^N(t) = |\{n \mid t \sqsubseteq_{\text{pre}} n, n \in N\}|$ for any token $t \in T$. The *positiveness* of a token t for any $t \in T$ is then defined as

$$f^P(t) = \frac{C^P(t)}{C^P(t) + C^N(t)}. \quad (2)$$

This represents the probability of a token $t \in T$ turning into a positive word.

A token $t \in T$ is considered to be *positive* if its *positiveness* is more than 50%, *negative* - the *positiveness* is less than 50%, and *neutral* - the *positiveness* is exactly 50%. We construct sets of *positive* tokens T^+, *negative* tokens T^-, and *neutral* tokens $T^=$, so that tokens can be re-weighted at each decoding step based on the desired sentiment. Simply put: we increase the value of the desired logits by adding weight before they are converted into probabilities by the softmax function. Figure 1 presents the described flow.

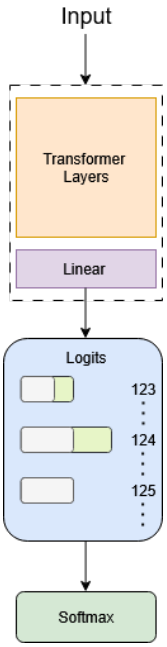


Figure 1: The value of logits is manipulated before the conversion.

The objective of the algorithm is to steer the model's output to the desired sentiment target by affecting token selection in the decoding process.

The algorithm alters the output logits before token selection, biasing it toward tokens that align with the desired emotional tone. Here the target is a more positive outcome, so the probability of selecting positive tokens is increased in two ways: an additive approach, described in Section 3.3, and the multiplicative approach described in Section 3.4.

3.1 Prompt dataset

To test our algorithm for three differently weighted cases and the baseline, we use an open-source instruction-following records dataset, databricks-dolly-15k [7] to prompt our model. The dataset consists of over 15,000 human-generated records with categories such as: *Creative Writing*, *Closed QA*, *Open QA*, *Summarization*, *Information Extraction*, *Classification* and *Brainstorming*. Categories *Closed QA*, *Open QA*, *Summarization*, *Information Extraction* and *Classification* are not suitable for our approach as their output adheres to the sentiment of the initial prompt by construction. Thus, in our experiment we use two categories from the dataset, *Creative Writing* and *Brainstorming*. The two categories are picked so that the model is not limited by the correctness of the answer from the start, but rather allow the maximum expression freedom. *Creative Writing* category contains 709 prompts and *Brainstorming* 1766.

3.2 Experimental setup

Open-source models allow direct access to the tokenizer and the output logits, which is necessary for our logit reweighting approach. We choose the Llama-3.1-8B-Instruct model, setting the decoding parameters based on the example provided by Meta [19]: $temperature = 0.6$ and $top_p = 0.9$. We keep $top_k = 50$, as specified in the Hugging Face transformers library [13].

Each prompt is run on different condition groups, using *apply_chat_template* from the Hugging Face transformers library [12] to format the input together with `add_special_tokens=False` to avoid duplicating special tokens. The template omits the `"role": "system"` field, to avoid sentiment biases before logit manipulation. Human evaluation sets the conditions on the output. Short text is read more carefully than long text, which the evaluator skims. The text also has to end on a sentence boundary. Generation therefore stops either at `eos_token_id` or, once the maximum of 300 tokens is reached, at the next end of sentence punctuation mark (!?.). A fixed random seed (1234) ensures reproducibility across all runs. The generation is run on A10G GPU. After the generation the outputs are evaluated with VADER (Valence Aware Dictionary and Sentiment Reasoner) a lexicon and rule-based sentiment analysis tool [14]. The VADER handles the slang and other informalities that our prompts might produce. For each output, VADER calculates the compound score, which has a value between −1, which is the most negative, and +1, which is the most positive. The sentiment of the text is considered *positive* if the compound score is greater than or equal to 0.05. The sentiment of the text is considered *neutral* if the compound score is between -0.05 and 0.05 and negative if the compound score is less than or equal to -0.05 [14].

3.3 Additive approach

Our first approach follows the principle of OpenAI's *logit_bias* parameter [21]. Manipulation of the logits is done by applying

an additive bias to the logits of positive sentiment tokens T^+ at every generation step. The additive approach applies the same magnitude of steering for every token in the selected set. Logits are modified at each decoding step:

$$\ell_i' = \ell_i + (w - 1), \quad \forall i \in T^+ \tag{3}$$

where ℓ_i is the logit of token i, w is the manipulation weight. Additive bias is chosen, so a token, whether it has a positive or negative logit, would be boosted. The bias is defined as $(w - 1)$, so $w = 1$ is the baseline.

The outputs also have to be complete, and not end in the middle of a sentence: an incomplete text invites the evaluators to judge it on its incompleteness rather than on its the sentiment. Outputs that reach the token limit are therefore eliminated, and those ending with the *eos* token or at a sentence boundary are kept. To determine the range of weights with complete outputs, we conduct a preliminary study. Applying the additive formula to the logits, outputs are generated from 12 prompts, in the range from 1.0 to 2.0 in steps of 0.1. All of the outputs terminate within the token limit.

Since all of the weights produce complete outputs, the main experiment is conducted with weights that are equally spaced by 0.25, starting from the baseline. From each category, *creative writing* and *brainstorming* we pick the 100 most positive, 100 most negative and 100 random neutral prompts. After removing duplicate prompts, 99 negative prompts and 97 neutral prompts in the *creative writing* category are left. On each of the six groups (positive prompts, negative prompts, neutral prompts of the two categories) weights $w \in \{1.0, 1.25, 1.5, 1.75\}$ are applied to the additive formula and steer in a positive direction.

The outputs are tested with Shapiro-Wilk tests, which show non-normality in each group. We use the Friedman test to compare the repeated measurements at different weights for the same prompts.Wilcoxon signed-rank tests with Bonferroni correction are also used for pairwise comparisons to the baseline and Kendall's W for the effect size. Five condition groups show no significant difference across weights, with p ranging from 0.7 to 0.88 and Kendall's $W < 0.005$. One condition group (category "Brainstorming", 100 most positive prompts) shows $p = 0.04$ but with a weak effect ($W = 0.028$) and no significance after the Bonferroni correction. In this group, the steering shifts the sentiment in the negative direction even though we intend to steer into the positive direction. After the correction, no pairwise comparison with the baseline reaches significance in any group, with the lowest corrected $p = 0.31$.

We raise weights to $w \in \{1.0, 2.0, 3.0, 5.0\}$, checking whether previous weights were too small. The Friedman test on a small sample size of 24 prompts (one duplicate instruction removed) does not show a significant difference ($p = 0.09$). The bias moves 0.5 to 1.3 extra tokens into the top-k set, while only 0.09 to 0.3 of them reach the top-p. Based on this, we decide that rather than testing an additive approach for steering into the negative direction, we try a different, more aggressive approach.

3.4 Multiplication/division of logits

An alternative manipulation of the logits is explored to achieve a stronger boost. Positive logits are boosted by multiplying them by different weights, and negative tokens by dividing them by the same weights. Our approach mirrors the work of Hollows [10], which uses a similar approach for penalizing repetition logits. The multiplication of logits does not apply the same magnitude of steering to each token in the selected set. A token with a logit further from zero will move further than the token closer to zero. The multiplication/division formula makes the manipulation depend on the arbitrary zero-point of the logits [10]. The arbitrariness of the zero-point of logits comes from the softmax function, which depends only on the differences between logits. Shifting all of the logits by the same constant produces the same output, which means the steering strength is dependent on where the model's logits lie.

The range of weights is established based on the completeness of the output text. Because the steering of the logits by multiplication is larger than with the additive formula, the upper weight is set to $w = 1.25$. The range between the baseline and upper weight in steps of 0.05 is tested, that is $w \in \{1.0, 1.05, 1.10, 1.15, 1.20, 1.25\}$. We generate outputs for 25 prompts at each weight. All the outputs generated with weights from $w = 1$ to $w = 1.20$ terminate within the token limit, while at $w = 1.25$ 24% of the outputs reach the token limit.

This upper limit is kept for the main experiment and the outputs that reach the token limit are filtered out for the human evaluation. The main experiment uses weights $w \in \{1.0, 1.05, 1.1, 1.15, 1.20, 1.25\}$, where $w = 1$ is a baseline and $w = 1.25$ is the upper limit. In each case, the function with the set of weights is applied to the logits in each decoding step. Each weight has a different steering magnitude, and unlike the additive formula, it scales differently between each token.

Since it is hard to evaluate sentiment from the text that does not form full sentences, the prompts are filtered. Prompts are removed if they begin with "Give me a list" or their response in the dataset contains a list, has four or fewer words, or has no verb. We take 60 prompts with the highest VADER score from the filtered prompts, 60 with the lowest VADER score, and 60 random prompts with a neutral score. Combining the two categories within each sentiment, gives 120 prompts for each sentiment group. A fixed random seed of 2025 is used.

The Friedman test shows a significant difference across weights in all of the condition groups, with $p < 0.001$ after the Bonferroni correction. Effect sizes for steering into the positive direction range between 0.255 and 0.267, while effect sizes for steering into the negative direction range between 0.081 and 0.154. A similar steering asymmetry is observed in the work of An et al. [1], who report model resistance to toxic steering in an instruction-tuned diffusion model, even though the steering goal, model type and method differ from ours. The pairwise comparisons against baseline reach significance at $w = 1.1$ in 4 groups: positive and neutral prompts steered into the positive direction and positive and neutral prompts steered into the negative direction. The group with negative prompts steered into the positive direction reaches significance in pairwise comparison against the baseline at $w = 1.05$ and the group with negative prompts steered into the negative direction reaches significance in pairwise comparison against the baseline at $w = 1.15$. All significant comparisons shift the sentiment into the intended direction. For example, steering into the negative sentiment with a weight of $w = 1.25$ for the negative-input prompt: "Write a short story (under 250 words) about meditation. The narrator should be uncomfortable." produces an output that ends with: "My mind refused to be stilled, a rebellious force refusing to be tamed by my feeble attempts at serenity." (VADER −0.98), while the last sentence for steering into positive sentiment is: " I might not have relaxed as promised, but at least we were free to go." (VADER +0.99).

4 Limitations

The main evaluation tool in this work is VADER, which determines the sentiment based on a sentiment lexicon. Our steering method is also lexicon-based, using the Hu and Liu lexicon with 32.6% of words also appearing in VADER's lexicon. This may result in some evaluation bias, so the results should be paired with a human evaluation that is yet to be conducted. The effect sizes are small and the multiplication/division formula can steer differently, based on the logits' position relative to zero. We do not test whether restricting the additive bias to a top-K candidate set would resolve the failure we diagnose. Results are limited to Llama-3.1-8B-Instruct.

5 Conclusion and Future Work

We test whether sentiment steering is possible with lexicon-driven logit manipulation on Llama-3.1-8B-Instruct, under two different logit manipulations: an additive and a multiplication/division approach. The additive approach shows no significant effect, while the multiplication/division approach shows significant bidirectional steering in all of the condition groups. Future work should test the steering approaches on different models and examine the effect based on the positioning.

References

[1] Hyeseon An and Yo-Sub Han. 2026. Dlm-swai: steering diffusion language models before they unmask. *arXiv preprint arXiv:2605.29626*.

[2] Hyeseon An, Shinwoo Park, Hyundong Jin, and Yo-Sub Han. 2026. Steering language models before they speak: logit-level interventions. *arXiv preprint arXiv:2601.10960*.

[3] Lekha Athota, Vinod Kumar Shukla, Nitin Pandey, and Ajay Rana. 2020. Chatbot for healthcare system using artificial intelligence. In *2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)*. IEEE, 619–622.

[4] Antonin Brun, Ruying Liu, Aryan Shukla, Frances Watson, and Jonathan Gratch. 2025. Exploring emotion-sensitive llm-based conversational ai. *arXiv preprint arXiv:2502.08920*.

[5] Sijia Cheng, Wen-Yu Chang, and Yun-Nung Chen. 2025. Exploring personality-aware interactions in salesperson dialogue agents. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, 60–71.

[6] Xusen Cheng, Ying Bao, Alex Zarifis, Wankun Gong, and Jian Mou. 2022. Exploring consumers' response to text-based chatbots in e-commerce: the moderating role of task complexity and chatbot disclosure. *Internet Research*, 32, 2, 496–517.

[7] Mike Conover et al. 2023. Free dolly: introducing the world's first truly open instruction-tuned llm. Retrieved June 30, 2023 from https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

[8] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

[9] Gaurush Hiranandani, Haolun Wu, Subhojyoti Mukherjee, and Sanmi Koyejo. 2025. Logits are all we need to adapt closed models. *preprint arXiv:2502.06806*.

[10] Peter Hollows. 2026. Gauge dependence and structured-output corruption in sign-branched repetition penalties: measurements across models, inference stacks, and alternative repetition controls. *arXiv preprint arXiv:2607.09791*.

[11] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177.

[12] Hugging Face. 2026. Chat templates. https://huggingface.co/docs/transformers/en/chat_templating. Transformers documentation, Hugging Face. (2026). Retrieved July 29, 2026 from.

[13] Hugging Face. 2024. Generation. Retrieved July 19, 2026 from https://huggingface.co/docs/transformers/main_classes/text_generation.

[14] Clayton Hutto and Eric Gilbert. 2014. Vader: a parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media* number 1. Vol. 8, 216–225.

[15] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, 2668–2677.

[16] Lasha Labadze, Maya Grigolia, and Lela Machaidze. 2023. Role of ai chatbots in education: systematic literature review. *International journal of Educational Technology in Higher education*, 20, 1, 56.

[17] Xun Liang et al. 2024. Controllable text generation for large language models: a survey. *arXiv preprint arXiv:2408.12599*.

[18] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6691–6706.

[19] Meta AI. 2024. Llama3/example_chat_completion.py. https://github.com/meta-llama/llama3/blob/main/example_chat_completion.py. GitHub repository. Accessed: 2026-07-19. (2024).

[20] Victor Monafal, Lawrence Patterson, Augustus Petrov, and Nicholas Robertson. 2024. Optimizing positive content generation in prompt-based abstractive summarization with large language models.

[21] OpenAI. 2024. Using logit bias to alter token probability with the OpenAI API. https://help.openai.com/en/articles/5247780-using-logit-bias-to-alter-token-probability-with-the-openai-api. Accessed: September 22, 2026. OpenAI Help Center, (2024).

[22] Elitza Stoilova. 2021. Ai chatbots as a customer service and support tool. *ROBONOMICS: The Journal of the Automated Economy*, 2, 21–21.

[23] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

[24] Kevin Yang and Dan Klein. 2021. Fudge: controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3511–3535.

[25] Jeewoo Yun and Jungkun Park. 2022. The effects of chatbot service recovery with emotion words on customer satisfaction, repurchase intention, and positive word-of-mouth. *Frontiers in psychology*, 13, 922503.

[26] Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. 2025. Controlling large language models through concept activation vectors. *arXiv preprint arXiv:2501.05764*.

[27] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

From Label Agreement to Downstream Performance: Evaluating LLM-Generated Training Data

Dario Mammana
dario.mammana.4448@student.uu.se
Uppsala University
Uppsala, Sweden

Petra Kralj Novak*
novakpe@ceu.edu
Central European University
Vienna, Austria

Abstract

Large language models offer a scalable and inexpensive alternative to human annotation, but agreement with a small human-labeled prompt-development sample may not adequately reflect the quality of LLM-generated labels as training data. We investigate this problem in multilingual hate-speech classification using an established diamond-standard dataset of English, Italian, and Slovenian social-media texts with four ordinal classes. We use Claude Haiku 4.5 to generate an alternative set of training annotations, fine-tune equivalent XLM-R classifiers on the human and Claude labels, and evaluate them against held-out human annotations. The classifier trained on Claude labels performs worse across languages and metrics, with the largest differences in less frequent classes, suggesting that systematic annotation differences can propagate into downstream classifier behavior. Our results show that, for the low-cost model examined here, high agreement on a prompt-development sample does not guarantee equivalent downstream classifier performance, highlighting the need to evaluate low-cost LLM annotations beyond prompt-level agreement.

Keywords

LLM annotation, annotation agreement, downstream evaluation, multilingual hate-speech detection

1 Introduction

Constructing labeled training datasets is one of the most costly and time-consuming stages of supervised machine learning. Human annotation is especially demanding for subjective tasks, where multiple judgments may be required for each item and disagreement cannot always be resolved by identifying a single correct label. Large language models (LLMs) offer a potentially inexpensive and scalable alternative to employing human annotators. Recent studies have shown that LLMs can perform some text-annotation tasks with accuracy comparable to that of crowd workers and at a substantially lower cost [5, 12]. This raises a practical question: can human annotations be replaced by LLM-generated labels when constructing training data for downstream classifiers?

Existing evaluations frequently address this question by measuring agreement between LLM-generated and human-provided labels. This agreement is informative, but it does not directly measure the usefulness of the resulting training data. Agreement describes the similarity between annotation sources, but it does not show whether the observed differences materially affect a classifier trained on those annotations. High overall agreement may also obscure systematic differences in less frequent classes. The quality of LLM-generated training data should therefore also be evaluated through its effect on the downstream model trained from those labels.

Hate-speech detection provides a particularly demanding setting for this evaluation. Definitions of hate speech and related forms of inappropriate content vary across annotation schemes, platforms, and social contexts [1, 4, 9]. Individual judgments may depend on context and annotator-specific severity thresholds, making disagreement an inherent property of the data rather than merely annotation error [2, 7]. Agreement tends to characterize clearer cases, while different labels for the same item can indicate ambiguity or proximity to category boundaries. Retaining both judgments preserves this disagreement in the training data rather than resolving it to a single label. Multilingual settings introduce an additional dimension, as annotation patterns and LLM behavior may vary across languages.

In this paper, we evaluate the downstream performance trade-offs of replacing human annotations with a low-cost, zero-shot LLM annotation workflow for multilingual hate-speech classification. We use a previously established diamond-standard dataset and experimental setup [6], covering English, Italian, and Slovenian social-media texts labeled on an ordinal scale distinguishing appropriate, inappropriate, offensive, and violent content. We generate an alternative set of training annotations using Claude Haiku 4.5, fine-tune equivalent XLM-R classifiers on the human- and LLM-annotated data, and evaluate both against held-out human annotations. We approximate a realistic annotation workflow in which a small human-labeled sample is available for prompt development, but obtaining human labels for the complete training set is impractical. We evaluate the resulting LLM annotations at two levels. First, we examine agreement and class-level differences between human and LLM annotations of the training data. Second, we measure how the annotation source affects downstream classification performance. This distinction is central to our study: similarity to human annotations and usefulness as training supervision are related, but they are not equivalent criteria.

Our results provide preliminary evidence that the classifier trained on human annotations performs better than the classifier trained on LLM-generated labels in the examined setting. The performance difference varies across languages and classes and is most pronounced for the less frequent and more severe hate-speech categories. Rather than drawing general conclusions across all LLM families, our findings provide a benchmark for a low-cost model and demonstrate why prompt-level agreement alone is insufficient to guarantee downstream utility.

*Also with Jožef Stefan Institute, Ljubljana, Slovenia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IS 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
https://doi.org/10.70314/is.2026.sikdd.43

2 Related Work

Recent studies have investigated LLMs as inexpensive and scalable text annotators. Gilardi et al. [5] report that ChatGPT outperformed crowd workers on several annotation tasks, while Ollion et al. [12] find that its suitability depends on the task and intended use. Results are less encouraging for subjective and sensitive classification problems, including hate speech [11]. Human–LLM collaborative annotation has therefore also been explored as an alternative to complete replacement [14].

A separate line of research treats annotator disagreement as meaningful information rather than annotation error. This perspective is particularly relevant to hate-speech classification, where judgments depend on context and individual severity thresholds. Kralj Novak et al. [7] retained two annotations per item when training monolingual classifiers. Grigor et al. [6] subsequently pooled the English, Italian, and Slovenian diamond-standard data to train and evaluate multilingual classifiers. The present study uses this disagreement-aware dataset and modeling setup to compare human and LLM annotation as alternative sources of training labels.

3 Methods

We simulate a realistic annotation workflow in which a small human-labeled sample is used for prompt development, while the remaining training data are annotated by the LLM. Human and Claude annotations are then used to fine-tune otherwise identical XLM-R classifiers, which are evaluated against the same held-out human-annotated test set. Krippendorff's ordinal alpha is used throughout to compare agreement between human and Claude annotations, between model predictions, and between predictions and human test annotations (Figure 1).

3.1 Datasets with Human Annotations

The dataset, introduced by Kralj Novak et al. [7], comprises three language-specific subsets of roughly 53,000 unique items each: YouTube comments in English and Italian and Twitter posts in Slovenian. Each item was labeled independently by two human annotators following the IMSyPP annotation guidelines[1] using four ordinal categories: 0–*appropriate*, 1–*inappropriate*, 2–*offensive*, and 3–*violent*. Inter-annotator agreement, measured using Krippendorff's ordinal alpha, is reported in Figure 1.

By design, the class distributions differ substantially between the training and test data. The training data were selected using an optimized sampling-before-labeling strategy [13] designed to increase the representation of minority hate-speech classes. The test data, by contrast, constitute a temporal holdout collected independently during a later period and closely reflect the natural class distribution [7].

3.2 Prompt Engineering and LLM Annotations

We use Anthropic's `claude-haiku-4-5-20251001` to generate an LLM-annotated counterpart to the complete human-annotated training set. Both training sets contain the same multilingual texts and differ only in their annotation source.

The prompt-development sample contains 150 human-annotated items per language, selected from the training set, and includes items on which the annotators disagree. Each ordinal category occurs at least 15 times across the two human annotations. Inter-annotator agreement in the sample is approximately the same as in the complete training set.

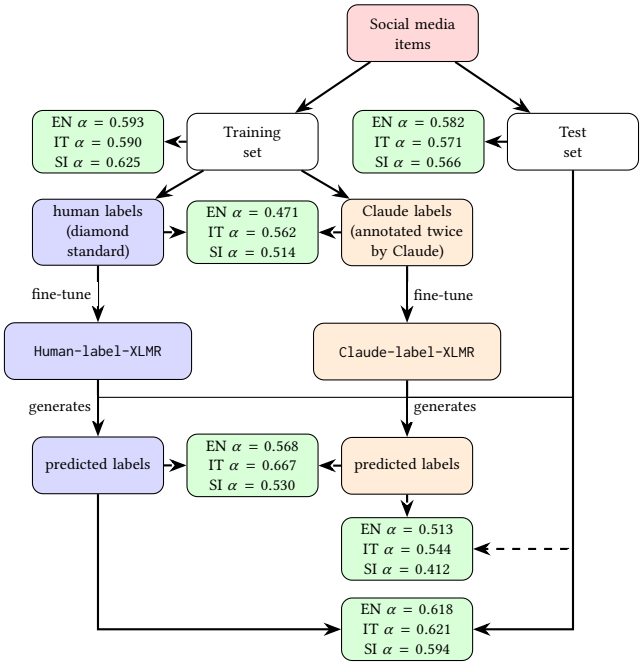


Figure 1: Pipeline overview. Green cells report Krippendorff's ordinal α between connected annotation sources or predictions. They summarize agreement across the experiment: between human annotators, human and Claude annotations, the two classifiers, and each classifier and human test annotations.

The LLM annotation procedure mirrors the diamond-standard structure of the human data by producing two labels for every item. This design is motivated by evidence that annotator disagreement provides useful supervision: agreement identifies relatively clear cases, while different labels indicate uncertainty near class boundaries [6]. Using two Claude labels therefore allows us to examine whether LLM annotations can provide both human-like judgments and comparable diversity in the training signal.

We iteratively refined the prompts on the English prompt-development sample, aiming to balance agreement with human judgments, measured by macro-F1, and diversity between the two Claude annotations, measured by Krippendorff's ordinal alpha (Table 1). Starting from the human annotation instructions, we first varied the temperature, but even the maximum temperature produced near-perfect within-Claude agreement ($\alpha = 0.99$). We therefore introduced strict and lenient annotator personas to induce variation in annotation thresholds while keeping the guidelines fixed. In total, we evaluated five configurations varying temperature, persona definitions, and task instructions. Across all configurations, the most prominent misclassification pattern involved the two most frequent classes, *appropriate* and *offensive*. The prompts and confusion matrices for all configurations, together with the code and experimental setup, are available in the project's GitHub repository.[2]

[1] Annotation guidelines: http://imsypp.ijs.si/wp-content/uploads/2021/12/IMSyPP_D2.2_multilingual-dataset.pdf

[2] Project GitHub repository: https://github.com/dariomammana/llm_labels_for_hate_speech

Table 1: Prompt development results across languages. α denotes Krippendorff's ordinal alpha; macro-F1 is averaged across all four classes.

Language	LLM α	Human α	Human-LLM α	Macro-F1
EN	0.878	0.593	0.615	0.595
IT	0.887	0.590	0.693	0.691
SI	0.912	0.625	0.618	0.533

Table 2: Label distribution of the two annotation regimes.

Source	Appropriate	Inappropriate	Offensive	Violent
Human	189,674 (59.27%)	10,960 (3.42%)	112,289 (35.09%)	7,111 (2.22%)
Claude	193,765 (60.56%)	28,138 (8.80%)	92,011 (28.76%)	6,017 (1.88%)

The prompts for the two strongest persona-based configurations were then translated into Italian and Slovenian and evaluated on the corresponding development sets. Their consistent behavior across languages supported the selection of a single configuration for all three training sets. Table 1 reports the results for the selected configuration.

Applying the selected configuration produced the class distributions shown in Table 2. Compared to the human annotations, Claude assigned more items to the *inappropriate* class and fewer to the *offensive* and *violent* classes. Thus, similar overall proportions of *appropriate* content conceal systematic differences among the less frequent categories.

3.3 Model Fine-tuning and Evaluation

We fine-tune two classifiers from the same pretrained XLM-R model [3]: `Human-label-XLMR` on the human annotations and `Claude-label-XLMR` on the Claude annotations. Rather than evaluating isolated models, we compare two realistic training pipelines. The architecture, data partitions, and base hyperparameters are held constant, while dataset-dependent elements such as class loss weights and validation checkpoint selection are computed separately within each pipeline.

We partition the training set into training (80%) and validation (20%) sets, stratified by language and label. Both labels for an item always occur in the same partition, thereby preventing data leakage between the training and validation sets.

Both classifiers are trained for three epochs using AdamW, a learning rate of 6×10^{-6}, and a batch size of 8. To account for class imbalance, we use weighted cross-entropy with weights inversely proportional to class frequencies in the respective training set. For each classifier, we select the checkpoint with the highest validation macro-F1. Fine-tuning was conducted on an NVIDIA A100 GPU.

Each test item has two human annotations, which are treated as two evaluation instances. The model therefore produces one prediction per unique item, and this prediction is compared separately with each of the two human judgments.

We report three complementary metrics. Accuracy is the proportion of model–annotation pairs with identical labels. Macro-F1 is the unweighted mean of the four class-specific F1 scores and therefore gives equal importance to frequent and infrequent classes. Krippendorff's ordinal alpha [8] measures agreement beyond chance while accounting for the ordered severity scale: disagreement between *appropriate* and *violent*, for example, is penalized more strongly than disagreement between adjacent categories. Following Kralj Novak et al. [7] and Grigor et al. [6], ordinal alpha is the primary metric because it accounts for chance agreement and permits comparison between model–annotator and inter-annotator agreement.

4 Results

`Human-label-XLMR` outperforms `Claude-label-XLMR` on all metrics and in all three languages (Table 3), with the smallest difference for Italian and the largest for Slovenian.

Table 3: Evaluation results for both models across all three languages. Alpha: Krippendorff's ordinal alpha. F1: macro-averaged F1 score.

Language	Ann. method	Alpha	Macro F1	Accuracy
EN	Human	**0.618**	**0.563**	**0.831**
	Claude	0.513	0.511	0.813
IT	Human	**0.621**	**0.663**	**0.842**
	Claude	0.544	0.580	0.785
SI	Human	**0.594**	**0.602**	**0.809**
	Claude	0.412	0.445	0.741

Figure 2 presents paired bootstrap estimates of the difference between the human-trained and Claude-trained models, with positive values favoring the former. All 95% confidence intervals exclude zero, and all comparisons remain statistically significant after Holm correction for multiple comparisons.

Table 4 reports the class-specific F1 scores for both models. The performance gap is negligible for the *appropriate* class but widens considerably for the hate-speech classes. It is largest for *violent* in Slovenian, where `Claude-label-XLMR` scores only 0.176, compared with 0.364 for `Human-label-XLMR`.

Table 4: Per-class F1 scores on the human-labeled test set. The best score for each language and class is shown in bold.

Lang.	Ann. method	Approp.	Inapprop.	Offensive	Violent
EN	Human	**0.888**	**0.325**	**0.713**	**0.327**
	Claude	**0.888**	0.280	0.584	0.292
IT	Human	**0.912**	**0.601**	**0.629**	**0.512**
	Claude	0.878	0.476	0.558	0.409
SI	Human	**0.866**	**0.469**	**0.710**	**0.364**
	Claude	0.843	0.264	0.498	0.176

5 Discussion

The results illustrate that prompt-level agreement is insufficient for evaluating training labels produced by a low-cost LLM. Although the selected prompt achieved reasonable agreement with human labels, the classifier trained on Claude labels performed consistently worse, with all comparisons remaining significant after Holm correction. Annotation agreement and downstream utility are therefore related but distinct evaluation criteria.

The class-level results suggest how annotation differences can propagate into downstream classifier behavior. Claude assigned a similar proportion of *appropriate* labels but more *inappropriate* and fewer *offensive* and *violent* labels than the human annotators (Table 2). Correspondingly, the difference between the classifiers

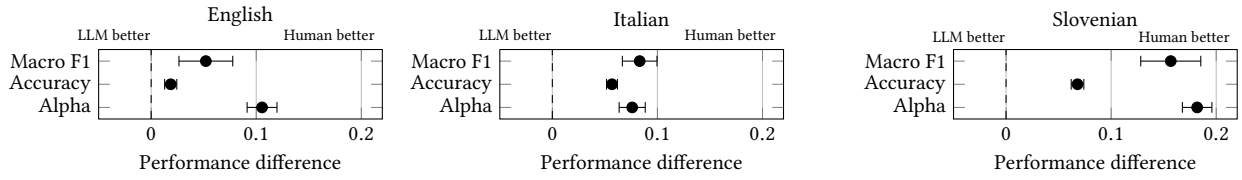


Figure 2: Paired bootstrap comparison between the human-trained and LLM-trained models. Dots denote the observed paired difference (human − LLM), and horizontal bars indicate the 95% bootstrap confidence intervals obtained from 5,000 paired resamples of test items with replacement. The dashed vertical line marks zero. All comparisons remain statistically significant after Holm correction for multiple comparisons.

was small for the *appropriate* class and larger for the less frequent classes, particularly *offensive* and *violent*. This pattern does not establish a causal relationship, but it indicates that relatively small systematic differences in training labels can produce substantial differences in class-specific performance.

Annotator diversity may also contribute to the observed performance gap. Despite persona prompting, agreement between the two Claude runs remained substantially higher than between human annotators. The Claude-annotated training data may therefore represent ambiguous cases less effectively than the human diamond standard. Future work should examine whether greater diversity can be obtained by combining annotations from different LLMs rather than eliciting two perspectives from the same model. Comparing single-LLM, multi-LLM, and human annotation regimes would help separate the effects of correspondence with human judgments from the value of disagreement itself. An earlier English-only experiment using a single LLM annotation per item similarly found that a classifier trained on LLM labels performed worse than one trained on human labels [10]. However, whether two LLM annotations per item provide better training supervision than one remains untested.

The economic advantage of LLM annotation remains substantial. Generating all 319,931 Claude labels, including the prompt-development experiments, cost less than USD 65 through the Anthropic API (June 2026). The present results therefore reveal a concrete trade-off: LLM annotation offers considerable savings and scalability, but these benefits were accompanied by lower downstream performance, especially for less frequent classes.

This study is limited to one small LLM (Claude Haiku 4.5), a single training run per model, and three languages. The larger performance gap observed for Slovenian may arise from the available language resources, the source platform of the data, or other annotation properties. Given the substantial cost advantage of this setup, future work should evaluate whether more capable models and practices such as chain-of-thought reasoning and systematic persona ablations can make agreement a more reliable proxy for downstream utility.

6 Conclusion

We evaluated LLM-generated training labels through both their agreement with human annotations and their effect on a downstream classifier. In this multilingual hate-speech experiment, XLM-R trained on Claude labels performed worse than XLM-R trained on human diamond-standard annotations, especially for less frequent classes. At the same time, the low annotation cost demonstrates the practical appeal of LLM-generated labels. These findings show that agreement on a small prompt-development sample may not adequately represent the downstream loss in quality. Robust protocols are therefore needed to estimate the utility of LLM-generated training labels from limited human-annotated data.

Acknowledgements

The authors acknowledge partial financial support from the Slovenian Research Agency (research core funding no. P2-103).

References

[1] Alexander Brown. 2017. What is Hate Speech? Part 2: Family Resemblances. *Law and Philosophy*, 36, 5, (Oct. 2017), 561–613. doi:10.1007/s10982-017-9300-x.

[2] Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a Perspectivist Turn in Ground Truthing for Predictive Computing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 6, (June 26, 2023), 6860–6868. doi:10.1609/aaai.v37i6.25840.

[3] Alexis Conneau et al. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, (July 2020), 8440–8451. doi:10.18653/v1/2020.acl-main.747.

[4] Paula Fortuna, Juan Soler, and Leo Wanner. [n. d.] Toxic, Hateful, Offensive or Abusive? What Are We Really Classifying? An Empirical Analysis of Hate Speech Datasets. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 6786–6794. Retrieved Apr. 13, 2026 from.

[5] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120, 30, (July 25, 2023), e2305016120. doi:10.1073/pnas.2305016120.

[6] Patricia-Carla Grigor, Bojan Evkoski, and Petra Kralj Novak. 2024. Multilingual Hate Speech Modeling by Leveraging Inter-Annotator Disagreement. In *Proceedings of Data Mining and Data Warehouses – SiKDD 2024*. Jožef Stefan Institute. doi:10.70314/is.2024.sikdd.7.

[7] Petra Kralj Novak, Teresa Scantamburlo, Andraž Pelicon, Matteo Cinelli, Igor Mozetič, and Fabiana Zollo. 2022. Handling Disagreement in Hate Speech Modelling. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer International Publishing, Cham, 681–695. doi:10.1007/978-3-031-08974-9_54.

[8] Klaus Krippendorff. 2019. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications, Inc. doi:10.4135/9781071878781.

[9] Seul Lee and Anne Gilliland. 2024. Evolving Definitions of Hate Speech: The Impact of a Lack of Standardized Definitions. In *Wisdom, Well-Being, Win-Win*. Springer Nature Switzerland, Cham, 141–156. doi:10.1007/978-3-031-57860-1_11.

[10] Dario Mammana. 2026. *Human vs LLM labels for fine tuning hate speech classification: a cost-effective comparative analysis*. Bachelor's Thesis. Central European University, Vienna, Austria.

[11] Anders Giovanni Møller, Arianna Pera, Jacob Dalsgaard, and Luca Aiello. 2024. The Parrot Dilemma: Human-Labeled vs. LLM-augmented Data in Classification Tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, St. Julian's, Malta, (Mar. 2024), 179–192. doi:10.18653/v1/2024.eacl-short.17.

[12] Etienne Ollion, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2023. ChatGPT for Text Annotation? Mind the Hype! (Oct. 4, 2023). Pre-published.

[13] Andraž Pelicon, Syrielle Montariol, and Petra Kralj Novak. 2023. Don't start your data labeling from scratch: OpSaLa-optimized data sampling before labeling. In *International Symposium on Intelligent Data Analysis*. Springer, 353–365.

[14] Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (CHI '24). Association for Computing Machinery, New York, NY, USA, (May 11, 2024), 1–21. ISBN: 979-8-4007-0330-0. doi:10.1145/3613904.3641960.

Short Term Time-Series Forecasting Framework

Gašper Kreft
gasper.kreft@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Aljaž Osojnik
aljaz.osojnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Martin Žnidaršič
martin.znidarsic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Bernard Ženko
bernard.zenko@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

This paper presents a configurable and easy to use framework for short-term time-series forecasting that combines automated feature extraction from time-series and exogenous variables, direct multi-horizon model training and their evaluation. The framework includes standard machine learning algorithms and is designed for quick offline model prototype development and real-time prediction, rather than development of state-of-the-art models and their fine tuning. We demonstrate its practical use for forecasting electricity consumption of a commercial gas station. Its modular design allows the forecasting pipeline to be easily adapted to different time-series variables and external data sources and enables quick prototyping of forecasting models.

Keywords

short-term forecasting, machine learning, timeseries forecasting framework, direct multi-horizon forecasting, tsfresh

1 Introduction

Short-term forecasting is frequently used in energy management systems to support the planning and operation of energy resources, and is the key component of the *smart grid*. Individual installations may combine conventional loads with technologies such as heat pumps, water storage tanks, battery storage, and controllable HVAC systems. These components can produce supply and demand patterns that vary over time, and their forecasting enables better resource planning and operation.

Forecasting models for real-world applications face several practical constraints. Temporal patterns can occur at different time scales, and may depend on different external or *exogenous* variables such as weather or number of customers. A model intended for operational use must also be trained and evaluated in a manner that mimics its practical use, and we must ensure that evaluation procedure only uses information that would have been available at the prediction time. A forecasting approach therefore needs to combine multiple feature sources while allowing the model configuration to be adapted to different forecasting tasks.

This work presents a configurable short-term forecasting framework designed around these requirements. The framework supports automated feature engineering, direct multi-horizon prediction, integration of exogenous variables and their forecasts, and real-time inference within a common pipeline. Its application is demonstrated on a real-life problem of forecasting electricity consumption of a commercial gas station. The work was performed within the SEEDS Project[1], where it represents a component of an aggregation platform for providing electricity flexibility services.

The source code and example data used to demonstrate the framework are publicly available at https://github.com/gkreft01/ts_framework/.

2 Related Work

Recent developments in time-series forecasting have increasingly demonstrated the effectiveness of machine-learning methods for complex forecasting tasks. The M5 Accuracy competition at Kaggle [3] provides strong empirical evidence in this direction. Across the competition, which involved 42,840 hierarchical retail sales time-series and related exogenous variables, the top-performing methods were all based on machine learning. In particular, the 50 highest-ranked methods outperformed the strongest statistical benchmark, with LightGBM-based approaches achieving substantial improvements in forecasting accuracy. The authors consequently identified the superior accuracy of machine learning methods and the value of exogenous variables as two of the main findings of the competition [3].

Automated feature engineering provides another approach for improving machine learning based time-series forecasting. Wang et al. [6] investigated the use of automatically extracted time-series features within an automated machine learning framework for multi-step-ahead forecasting. Their approach constructs a large set of statistical time-series features using the *tsfresh* library and subsequently applies machine learning models to the resulting feature representation. The authors found that incorporating time-series features improved forecasting performance and that multi-output forecasting generally performed better than recursive forecasting under the same computational budget. Their work demonstrates the potential of automated feature engineering for multi-step forecasting; however, the framework focuses on features derived from the target time series and does not incorporate exogenous variables such as weather forecasts.

The usefulness of exogenous variables may also depend on the forecasting horizon. Perez Guillen and Milanović [2] investigated the influence of exogenous variables, including weather, economic activity, and demographic information, across different forecasting time scales using LSTM models and real UK electric grid data. Their results identified an inflection point between approximately 30 and 90 days, below which the inclusion of exogenous variables could reduce forecasting accuracy, while beyond this range they improved it.

[1] https://project-seeds.eu/

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.100

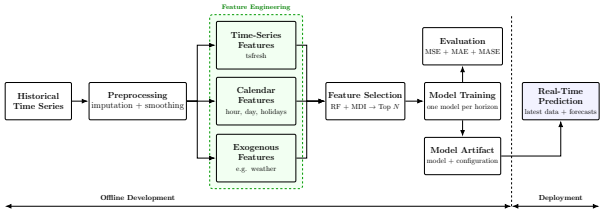


Figure 1: High-level overview of the proposed forecasting framework, showing different steps and components

The strategy used to generate multiple future predictions is an important consideration. Recursive forecasting trains a single one-step-ahead model and repeatedly feeds its predictions back into the model, which can lead to accumulated forecasting errors as the horizon increases. In contrast, direct forecasting trains separate models for the individual forecasting horizons, avoiding the propagation of previous prediction errors. Comparative studies in energy-demand forecasting and other real-world multi-step forecasting tasks have investigated these strategies using classic machine learning and deep learning models, demonstrating that the choice of forecasting strategy can substantially affect predictive accuracy, with direct and multiple-output approaches often outperforming recursive ones [4, 5].

Building on these findings, the proposed framework combines automated time-series feature extraction and selection, model training for direct multi-horizon forecasting, and model evaluation. It also enables easy deployment of the trained models. In contrast to approaches that rely only on information from the forecasted time-series, the framework is designed to also enable inclusion of exogenous variables such as weather. As such it is a useful tool for quick forecasting model prototyping in a range of time-series applications.

The work most similar to our is the AutoML approach for time-series proposed by [6]. Their pipeline is very similar to ours, but they use automatic search for all relevant hyperparameters, including window size, feature subset, learning algorithm and its hyperparameters. This approach is more complex and much more computationally expensive. It is therefore less appropriate for quick prototyping, though it is expected to produce more accurate forecasting models.

The following section describes the proposed framework, while Section 4 demonstrates its practical application. We conclude the paper by discussing insights gained from real-world implementation and suggesting directions for future work.

3 Methodology

The proposed forecasting framework is a configurable pipeline for multivariate time series forecasting and is implemented in *Python*. It consists of four stages: (1) data acquisition, (2) feature engineering, (3) feature selection, and (4) model configuration and training and (5) model evaluation. All components are parameterized through a unified configuration object (described in Section 4.1), allowing the same pipeline to be reused for different forecasting tasks by modifying only the experiment configuration. The trained model can also be easily deployed for real-time prediction. See Figure 1 for a high-level diagram.

3.1 Data Acquisition

The framework provides a flexible data acquisition layer for obtaining data from different sources. The data acquisition layer is intentionally separated from the remaining components of the framework, allowing users to adapt the database connections and data file acquisition procedures to their own data sources without affecting other components. We assume the time-series data points to be equidistant with the same time step.

3.2 Feature Engineering

3.2.1 Preprocessing. Prior to feature extraction and model training, missing timestamps are identified by comparing the input time series with a regular time index created with the defined time step, i.e., time step present in the data. Missing observations are then imputed using forward filling, carrying the most recently available value forward to preserve the regular temporal structure required by subsequent processing steps. Because non-aggregated time-series often include many spikes, we optionally smooth the data before continuing with the next steps.

3.2.2 Time-Series Features. Input time-series are transformed into a collection of statistical features using the *rolling window* approach and the *tsfresh* library [1]. The framework supports the extraction of features over multiple temporal windows to capture patterns occurring at different time scales. For each time-series, rolling windows of configurable lengths are used to represent both short-term and longer-term temporal behavior. The resulting features can therefore capture recent dynamics as well as patterns that develop over longer periods.

Prior to feature computation, differencing and moving-average smoothing can optionally be applied. Time-series can additionally be aggregated to a lower temporal resolution before rolling windows are applied. If this option is selected, features are computed for each rolling window and subsequently aligned back to the original sampling frequency using forward filling.

3.2.3 Calendar Features. In many problem domains the forecasted value depends on time of day, day of week or season of the year. The framework automatically derives features representing hour, weekday, weekend and month from the last timestep available at the time of forecasting. Additionally, it uses the *holidays* library to count the number of days until the next workday.

3.2.4 Exogenous Features. Forecasted variables are frequently affected by exogenous processes such as weather, for which we typically have forecasts provided by weather services. It is possible to include such forecasts and other exogenous variables as features in our framework. Available forecasts are transformed so that every prediction timestamp has future forecast values available for each forecasting horizon. During both training and testing, the framework uses archived numerical weather forecasts rather than observed meteorological measurements. This ensures that the model is trained under the same information constraints encountered during operational deployment, where only forecast weather is available. While this generally results in higher prediction errors due to the uncertainty introduced by meteorological forecasts, it provides a more realistic estimate of real-world forecasting performance and avoids optimistic bias.

3.3 Feature Selection

Using the *tsfresh* library combined with different time window sizes can result in a large number of features, many of which might be irrelevant to our forecasting problem. As a next step we therefore perform feature selection with the *Random Forest* algorithm, which we selected due to its efficiency and robustness, but it is straightforward to replace it with any feature selection

algorithm implemented in the *scikit-learn* library. We apply the algorithm for each forecasting horizon separately to compute the *Mean Decrease in Impurity (MDI)*, which is an estimate of feature importance. These feature importance scores are then averaged across all forecasting horizons, and the specified number of features with the highest mean importance is selected for training the final model.

3.4 Forecasting Model Training

The framework implements the direct multi-horizon model training process and trains a separate forecasting model for each prediction horizon using the selected training features and the corresponding target values. Again, we use the *Random Forest* algorithm for model training, due to its efficiency and robustness, but it can easily be replaced by some other algorithm from the *scikit-learn* library. The resulting multi-horizon model is as a model artifact together with the selected feature names, configuration, and training time. This allows the trained model to be subsequently loaded and evaluated or deployed for forecasting.

3.5 Evaluation

The trained model is evaluated on a held-out test set that is not used during model training. Before evaluation, the features required by the model (i.e., selected features before model training) are computed for the testing data. The model generates predictions for the complete forecasting horizon, which are then compared with the corresponding true target values. Performance is assessed using *Mean Squared Error (MSE)* and *Mean Absolute Error (MAE)*. In addition, *Mean Absolute Scaled Error (MASE)* is calculated by scaling the model's MAE against the MAE of a naïve forecast based on the value from the corresponding time step of the previous day. Lower values of all three metrics indicate better forecasting performance.

See Figure 2 for an illustration of testing the trained models on a test set. Red dotted rectangles represent the data from which the features were calculated, and the orange squares the predictions. Each x_i from the test set is first compared with the model's prediction $\hat{x}_{i,1}$ made one time step in advance then with the prediction $\hat{x}_{i,2}$ made two time steps in advance, and so on. In this context, we get a separate error measure for each horizon:

$$\mathrm{MAE}_h = \frac{1}{N-2m+1}\sum_{i=m}^{N-m}|x_{i+h}-\hat{x}_{i+h,h}|$$

$$\mathrm{MAE}_{\mathrm{naive},h} = \frac{1}{N-2m+1}\sum_{i=m}^{N-m}|x_{i+h}-x_{i-H}|$$

$$\mathrm{MASE} = \sum_{h=1}^{H}\frac{MAE_h}{MAE_{\mathrm{naive},h}},$$

where m is the size of the largest rolling window used for feature computation, N the test set size, and H the total number of horizons.

Finally, plots presenting predicted and true values on the test data are generated to provide a visual assessment of the model's forecasting behaviour.

3.6 Real-Time Prediction

The trained model can easily be deployed for real-time forecasting. The framework loads a previously trained model and retrieves the most recent data required for feature generation. The required historical period is determined automatically from the largest temporal window used by the model. Features are then generated using the same preprocessing and feature engineering procedures applied during training, and the most recent feature vector is passed to the trained model together with any required exogenous variables of forecasts. The model produces predictions for the configured forecasting horizon, which are assigned to the corresponding future timestamps. The resulting forecasts can then be stored in a database for subsequent use. This process allows the same trained model and feature engineering configuration to be used for operational forecasting.

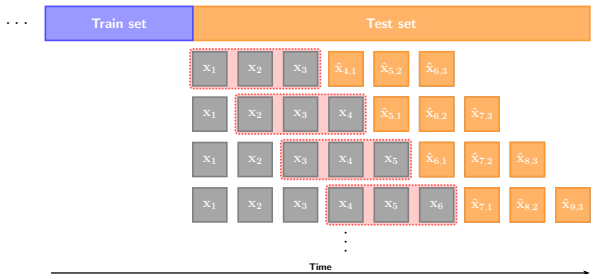


Figure 2: Illustration of the multi-horizon evaluation procedure. For each forecast origin, the red dotted rectangle denotes the measurements used for feature calculation, while the orange squares denote the resulting forecasts

4 Use Case Example

We demonstrate our tool on a task of forecasting electricity consumption of a single heat pump, which is a part of a pilot study within the SEEDS project. The data was collected by our project partner Petrol d.d., and comes from one of their commercial gas stations, while the weather data was collected from the Slovenian Environment Agency Web site. The use case scenario requires a forecasting model that generates one-day-ahead electricity consumption forecasts for a complete day in 15 min steps.

The example data set is provided in our repository (https://github.com/gkreft01/ts_framework/) as an SQLite database (file `example.db`). The database includes two tables, one with (normalized) heat pump electricity consumption (variable `Active_Power`) in 15 min steps for two months (ranging from 2026-03-01, 00:00h to 2026-05-01, 00:00h), and another with weather forecasts in 3 hour steps for the closest location (Jože Pučnik Airport), covering the same time interval. Weather data includes variables such as temperature, wind speed and precipitation.

4.1 Model Configuration and Training

The user needs to provide model configuration, which includes definition of the target variable, feature-extraction parameters, forecasting horizon, test period, model parameters, and optional external forecasts, These are defined through a `ModelConfig` object in file `experiments.py`.

In our example (see Listing 1), the target is `Active_Power`, sampled at 15 min steps, to which smoothing is applied with a window size 8 (`target_smooth_size`). Computation of time-series features is defined through method `FeatureSpec`, which specifies the rolling-window size, data resolution reduction (aggregation), generated feature set, output smoothing and differencing. In this configuration, two sets of features are computed, one with

rolling window size 96 and another with size 96*7, corresponding to one day and one week. The first set uses data at original 15 min resolution, while the second aggregates original 15 min data points into one daily value (i.e., 96 15 min steps). The third parameter determines whether *tsfresh* computes a bigger (`EfficientFCParameters`) or smaller (`MinimalFCParameters`) set of features, both sets use the bigger set (value `False`). The next parameter enables smoothing of output target predictions (used in the first set, where values are smoothed over 8 time steps). The last parameter enables differencing of data, but is not used in neither set (set to 0 or missing, instead of 1). The 50 highest-ranked features are retained (`top_n_features`), and a Random Forest model with 100 trees is used for direct multi-horizon forecasting. Temperature and accumulated precipitation forecasts (variables `temperature_c` and `precip_accumulated_mm`) are additionally specified as exogenous forecasts. The test set is defined as the last 14 values of the complete data set (`test_size`).

Listing 1: Model configuration example

```
HEAT_PUMP_CONFIG = ModelConfig(
    config_name="heat_pump",
    target_name="Active_Power",
    target_smooth_size=8,
    feature_params={
        "Active_Power": [
            FeatureSpec(96, 1, False, 8, 0),
            FeatureSpec(96*7, 96, False)
        ]
    },
    top_n_features=50,
    model_type=RandomForestRegressor,
    test_size=14,
    model_kwargs={
        "n_estimators": 100,
        "max_features": "log2",
        "random_state": 42
    },
    additional_forecasts={
        "Letališče Jožeta Pučnika Ljubljana":
        ["temperature_c", "precip_accumulated_mm"]
    }
)
```

Once model configuration (`experiments.py`) and database connection (`data_config.py`) are provided, the complete pipeline can be executed and model trained using a single command:

```
python pipeline.py --start-date 2026-03-01 --end-date
      2026-05-01 --config HEAT_PUMP_CONFIG
```

Results are stored in `model_runs.jsonl`, together with the current configuration parameters. We can now generate forecasts with:

```
python predict_realtime.py --config-name 'heat_pump' --
      forecast-date 2026-04-29
```

Scheduled calls of this command can provide us with real-time forecasts according to the model configuration. The forecasts are stored in a separate table in the database (`example.db`).

4.2 Forecast Results

Figure 3 illustrates the forecasts generated by the framework for the first day of the test set. The blue and red lines show the true and forecast energy consumption, respectively, while the dotted line represents the temperature forecast used as an exogenous input to the model. On the test set, the model achieved an MSE of 0.22, an MAE of 0.38, and a MASE of 0.90.

5 Conclusions

This paper presented a configurable framework for quick prototyping and deploying short-term time-series forecasting models.

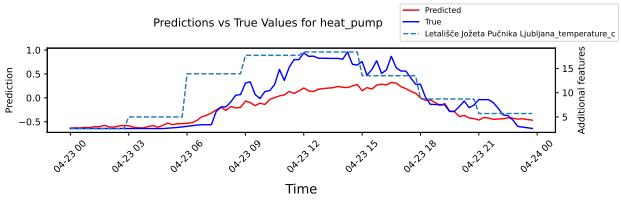


Figure 3: Example of heat pump forecasting for one day. The red line represents the predicted and the blue line true heat pump consumption; the dashed line is the temperature forecast used as an exogenous input

The framework combines automated feature extraction using *tsfresh* and feature selection, direct multi-horizon forecasting, and the integration of exogenous variables. Its modular architecture allows inclusion of different data sources, and simple adaptation to different forecasting applications.

The practical use case from the SEEDS project demonstrates how the tool can be easily configured and applied to a realistic forecasting scenario. The framework enables fast forecasting model prototyping and supports both offline evaluation and real-time prediction, using the same configuration.

6 Further Work

The current programmatic user interface can still be improved, but a suitable graphical user interface would further simplify configuration, model training, evaluation, and real-time forecasting, making the framework accessible to even wider audience.

Acknowledgments

The authors acknowledge the financial support from the European Union through the SEEDS project under the Horizon Europe Programme (GA No. 101138211) and the Slovenian Research and Innovation Agency (Research Core Funding No. P2-0103).

References

[1] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing* 307 (Sept. 2018), 72–77.

[2] Airam Perez Guillen and Jovica V. Milanović. 2025. The Influence of Exogenous Variables on Accuracy of Demand Forecasting over Different Time Scales. In *2025 IEEE PES/IAS PowerAfrica*. IEEE, Cairo, Egypt, 1–5. doi:10.1109/powerafrica65840.2025.11289083

[3] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting* 38, 4 (Oct. 2022), 1346–1364. doi:10.1016/j.ijforecast.2021.11.013

[4] Erik Fernando Mendez-Garces, David Buldain, and María Paz Comech. 2026. Comparative Evaluation of Direct and Recursive Multi-Step Forecasting for Electricity Demand Using Deep Learning and Gradient Boosting Models. *Energies* 19, 15 (July 2026), 3563. doi:10.3390/en19153563

[5] Souhaib Ben Taieb, Gianluca Bontempi, Amir Atiya, and Antti Sorjamaa. 2011. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. doi:10.48550/ARXIV.1108.3259

[6] Can Wang, Mitra Baratchi, Thomas Bäck, Holger H. Hoos, Steffen Limmer, and Markus Olhofer. 2022. Towards Time-Series Feature Engineering in Automated Machine Learning for Multi-Step-Ahead Forecasting. *Engineering Proceedings* 18, 1 (2022), 1–11. doi:10.3390/engproc2022018017

Short-Term Load Forecasting at the Device Level: Are specialized deep-learning approaches worth the effort?

Gal Zmazek
gz66721@student.uni-lj.si
Jožef Stefan Institute
Ljubljana, Slovenia

Aljaž Osojnik
aljaz.osojnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Martin Žnidaršič
martin.znidarsic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Bernard Ženko
bernard.zenko@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Short-horizon device-level load forecasting underpins demand-side management and local flexibility services, yet recent deep architectures have been validated almost exclusively on long-horizon benchmarks. We examine this gap on one device by adapting six architectures across five representative families—linear, hierarchical, mixer, patch-based, and exogenous-native—to a four-hour horizon at fifteen-minute resolution and evaluating them on a single commercial baseload meter with aligned weather channels, against classical baselines (persistence, linear regression, random forest, XGBoost). All models share one normalised dataset, one fixed temporal split, and one rescaling policy, isolating architecture as the variable. On this device we observe the following: first, performance across all deep architectures is tightly clustered, and a compact linear model comes within a few percentage points of the best deep architecture despite using orders of magnitude fewer parameters. Second, weather covariates benefit only those architectures purpose-built for exogenous fusion; among models with generic covariate support, identical weather inputs yield no measurable improvement or even degrade accuracy, and a capacity-matched control indicates that the effect is architectural rather than parametric.

Keywords

short-term load forecasting, device-level forecasting, exogenous covariates, weather, forecasting architectures, evaluation methodology

1 Introduction

Device-level load forecasting operates at short horizons: fifteen minutes to a few hours, consisting of only a few time steps. Deep-learning architectures for forecasting published over the last five years were developed and ranked predominantly on long-horizon benchmarks spanning hundreds of time steps over largely aggregate series. Whether their mechanisms still perform as intended at short horizons on data from a single (noisy) device is untested.

Weather appears as a natural exogenous driver of device load, however, the evidence for its impact is mixed. One study on heat-pump devices shows that transformers do not consistently beat XGBoost and tend to overfit on irregular series despite added capacity, with lagged load dominating feature importance and temperature having a negligible effect [13]. Exogenous support is also uneven across architectures: some are built around covariates [23]; several, including PatchTST [15], lack a mechanism for a second input series, so only the target's own history is consumed and incorporating weather requires modifying the architecture; while others accept covariates only over the forecast window, which presumes a weather forecast.

We explore whether recent architectures benefit from the additional information provided by historical weather measurements. We test six architectures: DLinear [25], TiDE [4], N-HiTS [2], TSMixerx [3], TimeXer [23] and PatchTST [15]; and six classical baselines: persistence, linear regression, an MLP, random forest and two XGBoost variants. We select data from one commercial meter under one persisted normalisation, with a fixed train/test split and window-rescaling policy. Each covariate-capable architecture is evaluated with and without additional past weather information.

Contributions: (i) a standardised way of comparing classical and recent architectures, training and evaluating every model under the same preprocessing and the same splits at 15 min, 1 h and 4 h horizons; (ii) a per-architecture weather comparison; and (iii) an empirical noise bound from repeated same-seed runs.

2 Related work

2.1 Classical and tree baselines

For load-forecasting, seasonal-naive baselines with calendar features remain the standard reference point [9]. For building-load forecasting, the field has shifted toward deep recurrent models [18], while tree ensembles have established themselves as a primary comparator. The supporting evidence is consistent: random forests were benchmarked against ARIMA, exponential smoothing, and neural networks [5] and achieve accuracy comparable to neural networks on building energy [1]. They are robust in their principal hyperparameter, such that an untuned forest is a fair comparator [24], and they retain an advantage at scale [14]. A gradient-boosted tree given identical window-based inputs matched eight deep architectures across nine datasets [6], and an inductive-bias advantage for tree models is well documented on medium-sized tabular problems [8]. In line with this, our tree baselines receive the same sliding-window inputs as the deep models, alongside the subsampled fit with which they are typically supplied.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/https://doi.org/10.70314/is.2026.sikdd.110

2.2 Deep-learning architectures

Recent load-forecasting research spans multiple families of deep-learning architectures. Attention-based forecasters [11] report accuracy gains that depend strongly on aggregation level [7, 20], which motivates evaluating them directly at device level. PatchTST segments a series into patches treated as tokens and encodes channels independently [15], typically with reversible instance normalisation [10]; its results are contested by linear models [25] and by architectures that attend across variates rather than along time [12]. Among prior device-level studies, the work closest to ours forecasts 308 heat pumps day-ahead and reports strong device heterogeneity [13].

2.3 Architectures that support historical weather

We found that exogenous support hinges on implementation rather than the published architecture, which commonly differ. Models exposing *historical* covariates can take past weather into account directly [2–4, 11, 17, 23], while models exposing only *future* covariates [19, 26] take a weather forecast as input; these fall outside our comparison, as we compare architectures against past weather only. Several architectures [12, 15, 22, 25] provide no covariate path.

3 Problem setup

Let y_t be the active power of one commercial meter at 15-minute cadence and $w_t \in \mathbb{R}^{17}$ the aligned weather vector. Given $x_t = (y_{t-L:t-1}, w_{t-L:t-1})$ with $L = 192$ (48 h), predict $y_{t:t+H-1}$ with $H = 16$ (4 h). Every trained model minimises squared error, so we report the matching quantity, RMSE in watts, at the single steps $h = 1$ (15 min), $h = 4$ (1 h) and $h = 16$ (4 h), and pooled across all 16 steps. Squared error weights the large deviations that a scheduler has to cover, and the persistence row in Table 1 anchors the scale.

The target is one always-on circuit of a service station, recorded from July 2020 to March 2024 (3.6 years): mean 618 W, standard deviation 104 W on the training slice, no zero readings, and a moderate seasonal swing (July mean 1.35 times February's). The test segment (June 2023 to March 2024) has a 10% lower mean. A heat pump would be a harder test of the weather question.

The series is split chronologically 80/20 into train-plus-validation and test, the last tenth of the training portion held out for early stopping. The test segment yields 24 864 windows, strictly later than everything the models see.

4 Methods

4.1 One pipeline

Recent architectures are usually benchmarked each under its own codebase and preprocessing, which leaves open whether a reported gain comes from the architecture or from the setup. To avoid that, every model in this paper, classical or deep, is trained and evaluated the same way: on the same normalised dataset, split into train and test at the same point in time, with every input window rescaled by the same declared rule, so that among the deep models a difference comes from the architecture rather than the data handling, and each weather comparison changes exactly one thing, the presence of the 17 weather channels. The classical baselines also differ in features and normalisation, so their gap is one between approaches. Per-column statistics are fitted on the training slice and shared by every run. Per-window instance normalisation centres and scales each lookback window by its own mean and standard deviation before the model sees it, then inverts that transformation on the output. This is the operation RevIN contributes to PatchTST [10] and that most recent architectures ship in some form; it is applied by the pipeline to the target channel of every deep model, with each model's internal version disabled so nothing normalises twice. Weather keeps the global normalisation only. RevIN's learnable affine parameters are not reproduced. Classical baselines receive the global normalisation alone.

4.2 Baselines and architectures

Persistence repeats y_{t-1}. Linear regression flattens the window. Random forest and XGBoost consume engineered features (load lags, rolling statistics, calendar encodings, weather at the forecast origin) and fit one regressor per horizon step. They are fitted twice: on a 15 000-window subsample, which is the usual practice, and on the same 89 357 windows the deep models see. An MLP flattens the raw window.

Recent architectures are taken from an existing implementation [16] through a thin adapter, so they train with the same loop, splits, normalisation and loss: DLinear, TiDE, N-HiTS, TSMixerx, TimeXer and PatchTST. Each receives the raw 192-step load window, a weather arm also the 17 weather channels. Library defaults apply, with a shared 25-epoch budget and early stopping. TimeXer runs at the shared 128-dimensional, 3-layer width rather than its own default of 512, which is four times every other Transformer here.

4.3 Constructing a weather path for PatchTST

PatchTST encodes each channel independently, so attention runs over the 24 time patches within a channel and never across them, and the head keeps only the target's forecast. Feeding weather as extra channels therefore does nothing: built from the same seed, the concatenated and univariate models have identical parameters and bit-identical output.

We therefore construct a weather path by hand. Both streams, load and the 17 weather channels collapsed into one combined signal, are built the same way up to the point where they meet: each is split into 24 overlapping patches, each patch is mapped to a 128-dimensional embedding by one linear layer plus a learned positional code, and the resulting sequence of embeddings is treated as a sequence of tokens and passed through its own three-layer self-attention encoder. This gives two independently-encoded token sequences, $Z_\ell \in \mathbb{R}^{24\times128}$ for load and $Z_w \in \mathbb{R}^{24\times128}$ for weather, each having so far attended only to itself. The two constructions differ only in how Z_ℓ and Z_w are then joined:

$$\text{positional:} \quad Z = \mathrm{W}[Z_\ell \,\|\, Z_w] \tag{1}$$

$$\text{attentional:} \quad Z = Z_\ell + \mathrm{MHA}(Q{=}Z_\ell,\ K{=}V{=}Z_w) \tag{2}$$

In the positional join, called the *separate encoder* model in Tables 1 and 2, $[\cdot\|\cdot]$ concatenates the two streams at each matching patch position into a 256-dimensional vector, and W is one learned linear layer projecting it back down to 128 dimensions. In the attentional join, MHA is multi-head cross-attention: the load tokens Z_ℓ act as the query Q, the weather tokens Z_w act as both the key and value ($K = V = Z_w$), so each load patch computes a weighted combination of all 24 weather patches instead of being tied to the matching one, letting a lag be learned rather

than specified. Neither construction is novel: cross-attention between an endogenous and an exogenous stream is published as TimeXer [23] and, with two encoders, as ExoTST [21].

5 Results

5.1 Model comparison

Because each model configuration is evaluated using a single random seed, a reported difference in test RMSE may reflect run-to-run variability rather than a genuine difference between models. To avoid overinterpreting small differences, we estimate an empirical reproducibility floor. We reran the linear model and the univariate PatchTST five times each while keeping the random seed, machine, data, and training code unchanged. For repeats whose early stopping terminated at the same epoch, the observed variation is attributable to low-level computational nondeterminism rather than to differences in model or data. The five linear repeats stopped at the same epoch and spanned 0.06 RMSE, with a standard deviation of 0.03. The PatchTST repeats that stopped at the same epoch differed by 0.07 RMSE in the same-epoch pairs we inspected. The largest same-epoch pairwise gap observed was 0.075 RMSE. We round this maximum up to 0.1 RMSE and use that value as a decision floor for comparisons on this dataset. This rule is conservative and operational, not inferential. An RMSE difference smaller than 0.1 is treated as within the reproducibility range of the single-seed evaluation protocol and is not interpreted as evidence that one model is better than another. An RMSE difference larger than 0.1 is reported as an apparent difference that exceeds the observed computational-noise floor; it is not claimed to be statistically significant.

Random forest and Random forest‡ are the identical model on the identical features (the last 48 load lags plus the weather reading at the forecast time, about 65 inputs); the plain row is fit on the usual 15 000-window subsample, the ‡ row on the full 89 357-window training set every deep model sees. XGBoost appears three times because we tried two feature sets and one data-size variant: XGBoost, calendar adds calendar encodings (hour-of-day and day-of-week as sine-cosine pairs, a weekend flag) to 48 load lags and weather, about 70 inputs; XGBoost, rolling stats uses 24 load lags, the mean, standard deviation, minimum and maximum of the load over the last 6, 12 and 24 hours, and the same weather reading, 53 inputs; and XGBoost, rolling stats‡ repeats the rolling-stats feature set on the full training set instead of the subsample.

Table 1 gives the raw 15-minute results. Considering only models without weather as input, the six explored architectures span just 1.1% (69.6 to 70.4), against a gap of up to 6.3% from the best of them to the best tabular baseline. Taking into account weather variations, the gaps go from 1.1% to 5.4% on the span across architectures and from 6.3% to 7.7% on the gap to the best tabular baseline, since weather helps some architectures and hurts others. DLinear reaches 70.3 with 6,176 parameters, within 2.4% of the best model overall, PatchTST + separate encoder, which is 144 times larger and correspondingly more expensive to train. TSMixerx at 48 K parameters matches univariate PatchTST within that 0.1 RMSE margin. More data does not close the tabular gap: on all 89 357 windows the random forest gets worse (77.0 to 78.1) and XGBoost moves by 0.2. PatchTST, library, is the implementation of [16] without the changes of Methods.

Table 1: Test RMSE (W), raw 15-minute target, single seed. † predates pipeline unification. ‡ fitted on all 89 357 training windows rather than a 15 000-window subsample.

Model	Params	Overall	15 min	1 h	4 h
Persistence	–	93.9	62.6	94.0	97.8
Random forest‡	–	78.1	61.9	77.0	81.9
Random forest	–	77.0	63.2	75.9	80.6
Linear	55 K	76.0	57.0	74.6	80.4
MLP†	955 K	75.6	71.5	73.6	77.9
XGBoost, calendar†	–	74.7	55.5	73.9	77.9
XGBoost, rolling stats†	–	74.2	55.1	73.0	78.4
XGBoost, rolling stats‡	–	74.0	54.1	72.9	78.5
N-HiTS + weather	6.2 M	72.4	56.4	72.0	75.0
N-HiTS	2.9 M	70.4	52.0	69.9	73.9
DLinear	6 176	70.3	52.9	69.6	73.8
TiDE	1.3 M	70.3	52.7	69.5	73.8
TiDE + weather	2.1 M	70.3	52.7	69.4	73.8
TSMixerx	48 K	69.7	52.9	69.1	73.3
TSMixerx + weather	50 K	69.7	55.2	69.3	72.7
PatchTST	452 K	69.6	52.3	68.7	73.2
TimeXer	650 K	69.6	52.6	69.0	72.9
PatchTST, 8 layers	1.1 M	69.5	52.0	68.9	73.0
PatchTST, library	452 K	69.4	51.9	69.1	72.7
TimeXer + weather	650 K	69.0	52.3	68.4	71.8
PatchTST + cross-attn	1.1 M	68.9	52.2	68.7	71.6
PatchTST + separate enc.	887 K	**68.7**	52.3	68.4	71.5

5.2 Weather

Table 2 pairs each architecture's weather arm (with weather as input) against its univariate arm (without). All native arms receive identical weather, so a difference in gain comes from how each architecture uses it, not from having different weather to work with. Three of the six, TimeXer and our two PatchTST constructions, were built specifically for exogenous fusion, and all three gain: the separate-encoder join by 1.37%, cross-attention by 1.10%, and TimeXer by 0.87%. The other three, TSMixerx, TiDE and N-HiTS, take weather through a general-purpose covariate mechanism rather than one designed for weather specifically; of these, TSMixerx and TiDE move by less than that 0.1 RMSE margin, and N-HiTS degrades by 2.9%.

Between the two PatchTST constructions, the simpler one wins: the separate-encoder join gains 1.37% at 887 K parameters, against 1.10% for cross-attention at 1.1 M. Its extra capability, a learned lag over the weather patches, does not pay off here.

Both constructions carry more parameters than the univariate model they are compared against, so we ran a capacity-matched control: univariate PatchTST at eight encoder layers, 1 114 264 parameters against cross-attention's 1 120 040. It scores 69.51. Multiplying the parameters of the univariate model by 2.5 therefore buys 0.11 of RMSE, 0.15%, right at that 0.1 RMSE margin, while the same budget spent on a weather path buys 0.66, and 0.85 for the positional join with fewer parameters than the control. The fusion effect is six to eight times the capacity effect, so the gains in Table 2 are weather, not size. TimeXer's 0.87% needed no control: exogenous series enter as variate tokens, so adding weather adds tokens rather than weights [23].

6 Discussion

On this device, long-horizon architectures transfer but do not differentiate. Without weather, mechanisms that separate them

Table 2: Weather arm against univariate arm. Negative favours weather.

Architecture	Path	Δ	Δ %	Params
PatchTST	separate encoder	−0.96	−1.37	2.0×
PatchTST	cross-attention	−0.76	−1.10	2.5×
TimeXer	native	−0.60	−0.87	unchanged
TSMixerx	native	−0.01	−0.02	+2 K
TiDE	native	−0.01	−0.01	+800 K
N-HiTS	native	+2.01	+2.85	+3.3 M

on long-horizon benchmarks compress into a 1.1% band at four hours, with a 6.3% gap to the best tabular model. At this horizon the choice among recent architectures matters less than the choice between the recent field and a tabular model.

On this device, weather helps only where the architecture was designed around it. Here a weather benefit is a property of a particular design rather than of the task, which is consistent with reading published fusion gains as architecture-specific and with the device heterogeneity reported by Lüdecke et al. [13]. Within the constructed designs the simpler join is the better one.

Our study also has some important shortcomings and limitations. We ran a single seed and no significance test on the test errors; same-seed repeats alone differ by up to 0.07 RMSE, on the order of several differences in Table 1. We use data of one meter at one site, so no claim about other devices or sites can be made. Library defaults were used rather than per-model tuning, and the PatchTST family carries a learning rate tuned in earlier work.

7 Conclusion

On the one commercial meter we studied, recent deep-learning architectures transfer to short-horizon load forecasting but do not provide meaningful performance differentiation; the gap between models is narrow, and a 6,176-parameter DLinear model comes within 2.4% of the best. Furthermore, on this meter weather improved accuracy only in architectures with a mechanism designed for exogenous fusion, so the gain depends on the design and not only on the data.

For this device, the complexity of the larger deep-learning models is not justified by their marginal accuracy gains over a lightweight one such as DLinear; whether this holds for other, more weather-driven devices remains to be tested.

Acknowledgements

The authors acknowledge the financial support from the European Union through the SEEDS project under the Horizon Europe Programme (GA No. 101138211) and the Slovenian Research and Innovation Agency (Research Core Funding No. P2-0103).

References

[1] Muhammad Waseem Ahmad, Monjur Mourshed, and Yacine Rezgui. 2017. Trees vs Neurons: Comparison between Random Forest and ANN for High-Resolution Prediction of Building Energy Consumption. *Energy and Buildings* 147 (2017), 77–89. doi:10.1016/j.enbuild.2017.04.038

[2] Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. 2023. N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 6989–6997. arXiv:2201.12886 doi:10.1609/aaai.v37i6.25854

[3] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. 2023. TSMixer: An All-MLP Architecture for Time Series Forecasting. *Transactions on Machine Learning Research (TMLR)* (2023). arXiv:2303.06053

[4] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Transactions on Machine Learning Research (TMLR)* (2023). arXiv:2304.08424 https://openreview.net/forum?id=pCbC3aQB5W

[5] Grzegorz Dudek. 2015. Short-Term Load Forecasting Using Random Forests. In *Intelligent Systems'2014*. Springer International Publishing. doi:10.1007/978-3-319-11310-4_71

[6] Shereen Elsayed, Daniela Thyssens, Ahmed Rashed, Hadi Samer Jomaa, and Lars Schmidt-Thieme. 2021. Do We Really Need Deep Learning Models for Time Series Forecasting? arXiv:2101.02118 [cs.LG]

[7] Elena Giacomazzi, Felix Haag, and Konstantin Hopf. 2023. Short-Term Electricity Load Forecasting Using the Temporal Fusion Transformer: Effect of Grid Hierarchies and Data Sources. In *Proceedings of the 14th ACM International Conference on Future Energy Systems (e-Energy)*. 353–360. doi:10.1145/3575813.3597345

[8] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?. In *Advances in Neural Information Processing Systems 35, Datasets and Benchmarks Track*.

[9] Rob J. Hyndman and George Athanasopoulos. 2021. *Forecasting: Principles and Practice* (3 ed.). OTexts.

[10] Taesung Kim, Jinhee Kim, Yoonbeen Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2022. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations (ICLR)*.

[11] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *International Journal of Forecasting* 37, 4 (2021), 1748–1764. doi:10.1016/j.ijforecast.2021.03.012

[12] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

[13] Marcel Lüdecke, Timon Justi, Michel Meinert, and Bernd Engel. 2026. Comparing Load-Forecasts of Residential Heatpumps with Transformer and XGB on Field Data. In *Energy Innovation Conference (EnInnov)*. doi:10.34657/33291

[14] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. M5 Accuracy Competition: Results, Findings, and Conclusions. *International Journal of Forecasting* 38, 4 (2022), 1346–1364. doi:10.1016/j.ijforecast.2021.11.013

[15] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*.

[16] Kin G. Olivares, Cristian Challú, Azul Garza, Max Mergenthaler Canseco, and Artur Dubrawski. 2022. NeuralForecast: User Friendly State-of-the-Art Neural Forecasting Models. PyCon Salt Lake City, Utah, US. https://github.com/Nixtla/neuralforecast

[17] Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. 2023. Neural Basis Expansion Analysis with Exogenous Variables: Forecasting Electricity Prices with NBEATSx. *International Journal of Forecasting* 39, 2 (2023), 884–900. doi:10.1016/j.ijforecast.2022.03.001

[18] Ioannis Patsakos, Eleni Vrochidou, and George A. Papakostas. 2022. A Survey on Deep Learning for Building Load Forecasting. *Mathematical Problems in Engineering* 2022 (2022), 1–25. doi:10.1155/2022/1008491

[19] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191. doi:10.1016/j.ijforecast.2019.07.001

[20] Ayas Shaqour, Tetsushi Ono, Aya Hagishima, and Hooman Farzaneh. 2022. Electrical Demand Aggregation Effects on the Performance of Deep Learning-Based Short-Term Load Forecasting of a Residential Building. *Energy and AI* 8 (2022), 100141. doi:10.1016/j.egyai.2022.100141

[21] Kshitij Tayal, Arvind Renganathan, Xiaowei Jia, Vipin Kumar, and Dan Lu. 2024. ExoTST: Exogenous-Aware Temporal Sequence Transformer for Time Series Prediction. In *IEEE International Conference on Data Mining (ICDM)*.

[22] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. 2024. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

[23] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. 2024. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper_files/paper/2024/hash/0113ef4642264adc2e6924a3cbbdf532-Abstract-Conference.html

[24] Zeyu Wang, Yueren Wang, Ruochen Zeng, Ravi S. Srinivasan, and Sherry Ahrentzen. 2018. Random Forest Based Hourly Building Energy Prediction. *Energy and Buildings* 171 (2018), 11–25. doi:10.1016/j.enbuild.2018.04.008

[25] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 11121–11128. doi:10.1609/aaai.v37i9.26317

[26] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11106–11115. arXiv:2012.07436

Data-Driven and Model-Based Causal Discovery in Sparse Multivariate Time Series: A Comparative Framework for Sustainable Development Systems

Joao Pita Costa†
IRCAI
Institut Jozef Stefan
Ljubljana
joao.pitacosta@ijs.si

Andrej Srakar
IRCAI
Institut Jozef Stefan
Ljubljana
andrej.srakar@ijs.si

Oleksandra Topal
Department of AI
Institut Jozef Stefan
Ljubljana
oleksandra.topal@ijs.si

Jan Šturm
Department of AI
Institut Jozef Stefan
Ljubljana
jan.sturm@ijs.si

Abstract / Povzetek

This paper investigates Sustainable Development Goal (SDG) indicators as sparse multivariate time series, focusing on the identification and interpretation of relationships between development dimensions over time. Three complementary approaches are explored: SHAP-based explainable machine learning to identify influential indicators, Transfer Entropy to assess directional information flows, and StreamStory to characterize temporal dynamics through multivariate states and transitions. The study investigates whether causal structures inferred from empirical time-series data can be validated through complementary analytical approaches and compared with model-derived causal structures. Particular attention is given to the degree of agreement between methods, as converging evidence may provide stronger support for inferred relationships and reveal previously unrecognized system dynamics. It resolves into a problem of SDG causal discovery under small-n, multivariate, nonlinear, and incomplete time-series conditions. The results contribute to a comparative framework for exploring, validating, and interpreting causal patterns in sparse multivariate development data, while highlighting the methodological challenges associated with causal discovery from short, incomplete, and highly interdependent time series.

Keywords / Ključne besede

SDG; Causal Discovery; Multivariate Time Series; Transfer Entropy; Explainable Artificial Intelligence.

1 Introduction

The UN SDGs provide a comprehensive framework for understanding global development through 17 interconnected goals and a large system of statistical indicators. These indicators cover diverse dimensions including poverty, health, education, infrastructure, economic development, inequality, and environmental sustainability. Data are collected and maintained by international organizations such as the United Nations and the World Bank, as well as aggregated repositories such as Our World in Data. Most SDG indicators are available as longitudinal observations across countries, providing an important empirical basis for studying development trajectories over time. However, the same characteristics that make these datasets valuable for monitoring also make them particularly challenging for causal analysis.

SDG indicators are typically observed at annual frequency, frequently contain missing observations, have heterogeneous temporal coverage, and are strongly interdependent across sectors and geographical contexts. These characteristics create a fundamental methodological problem for the direct application of causal discovery techniques [1]. In contrast to typical causal discovery, an SDG time series may contain only a few observations for a country, with substantial gaps between them and changes in measurement practices over time.

Traditional causal inference relies heavily on predefined structural causal models or expert-provided Directed Acyclic Graphs (DAGs) (Pearl, 2009). However, in complex, high-dimensional applications typically encountered in knowledge discovery and data mining, complete domain expertise is rarely available, making manual graph construction error-prone or impossible. Causal discovery bridges this fundamental gap by operating as a data-driven inverse problem; it automates the learning of the underlying causal skeleton and directions directly from observational or temporal datasets (Vowels et al., 2022; Zhang et al., 2025). Understanding its advantages is of great importance for causality research at present.

The problem is further complicated by the multivariate nature of sustainable development. SDG indicators do not evolve independently: economic, social, environmental, health, and infrastructure variables can influence one another, while common external factors can simultaneously affect multiple indicators. Development systems may also contain delayed effects and feedback mechanisms. The resulting time series can therefore exhibit nonlinear dependencies, temporal autocorrelation, common trends, and bidirectional relationships. The original analysis similarly identifies SDG systems as short, noisy, highly interdependent, nonlinear, and potentially characterized by feedback loops. These properties create opportunities for causal discovery, but also increase the risk of

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2025 Copyright held by the owner/author(s).
http://doi.org/ 10.70314/is.2026.sikdd.122

interpreting statistical dependence or temporal precedence as evidence of causation.

The present study builds on this perspective by treating the SDG indicator system as a collection of sparse multivariate time series and examining three complementary analytical approaches: SHAP for interpretable predictive influence, Transfer Entropy for directional temporal dependencies, and StreamStory for multivariate state transitions [4]. This provides a richer econometric and statistical overview about the relationships studied.

2 Data and Methods

Our framework is intended to provide a more cautious and technically grounded approach to causal discovery in sustainable development systems, where limited observations, missing data, nonlinear interactions, and feedback mechanisms constrain the application of conventional methods.

The dataset considered in this study spans 1987–2023, giving a maximum of approximately 289 indicators with annual observations [1] for a complete country-level series [2], mostly sourced at [5] and [6], while substantial portions of the underlying indicator matrix are incomplete. Consequently, methods designed for longer or denser time series cannot necessarily be transferred directly to this setting without examining their statistical stability and sensitivity to preprocessing choices. Missing data represent an additional and particularly important challenge. Gaps in SDG datasets are not necessarily random: reporting capacity, national statistical systems, indicator definitions, and geographical or socioeconomic characteristics can influence whether an observation is available. The resulting missingness can therefore introduce systematic biases rather than simply reducing sample size. Moreover, imputation is not neutral from a causal perspective. Reconstructing missing observations may smooth temporal trajectories, alter autocorrelation structures, or introduce relationships between variables that were not directly observed. The study therefore treats completeness, missingness patterns, and imputation as explicit components of the analytical pipeline rather than as a purely technical preprocessing step.

These limitations are particularly relevant when applying established causal-discovery methods. Granger causality provides a well-established framework for testing whether the past of one time series improves the prediction of another, but its conventional formulations are sensitive to assumptions concerning stationarity and model specification. The original study found that direct application of Granger causality was problematic for the SDG data because many series were non-stationary. Other causal discovery approaches, including Structural Equation Models (SEM), the PC and FCI algorithms, and LiNGAM, impose assumptions concerning linearity, acyclicity, independence, or stationarity that may not correspond well to complex development systems with feedback and nonlinear interactions. Information-theoretic approaches provide an alternative perspective. Transfer Entropy (TE), introduced by Schreiber, measures the directed information transferred from one time series to another while accounting for information contained in the target's own history[3]. Formally, it is introduced as:

$$TE_{X\to Y}=I\left(Y_{t+1};X_t^{(k)}|Y_t^{(l)}\right)$$

where the analysis estimates conditional mutual information I from short empirical sequences X and Y at a given fixed time t.

Unlike conventional correlation measures, TE can capture nonlinear and time-lagged dependencies without requiring a specific parametric model. The previous study therefore used TE to examine directional relationships between influential SDG indicators and target variables, including bidirectional comparisons to distinguish potential upstream and downstream relationships. However, the flexibility of TE does not eliminate the fundamental limitations imposed by short time series. Estimation of information-theoretic quantities can itself be sensitive to sample size, history length, estimator choice, noise, and preprocessing. In the previous implementation, for example, TE was calculated independently for each country using a fixed history length of four observations. This raises an important methodological question: to what extent can the resulting directional relationships be considered robust when the available empirical evidence is intrinsically limited?

A complementary perspective, building on the work published in [2] and [3] focusing SDG 1, is provided by explainable machine learning. SHAP (SHapley Additive exPlanations) can quantify the contribution of individual features to the predictions of a machine-learning model[3] (see Figure 2) and thereby provide an interpretable representation of predictive relevance. In the previous study, Random Forest models were used to reduce the original high-dimensional indicator space, after which SHAP was used to identify the most influential indicators for each target variable. This approach is useful because it distinguishes the question of which variables are informative for prediction from the more demanding question of which variables participate in directional temporal relationships. Importantly, the two notions need not coincide (see Figure 1).

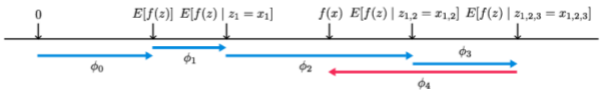


Figure 1: SHAP (SHapley Additive exPlanation) values attribute to each feature the change in the expected model prediction when conditioning on that feature (source: Lundberg and Lee, 2017).

The previous results already indicated that some indicators with high SHAP importance had relatively weak Transfer Entropy, whereas some indicators with moderate predictive importance exhibited stronger temporal information transfer. This divergence provides an important motivation for comparing complementary analytical approaches rather than relying on a single causal-discovery method.

[1] The detailed list of SDG indicator timeseries collected https://github.com/IRCAI-SDGobservatory/data/blob/main/IRCAI-SDGObservatory-Indicators.csv

[2] Data sources analyssis available at IRCAI?s SDG Observatory: https://sdg-observatory.ircai.org/index.php/sdg-16/).

[3] SDG application by IRCAI/E3 with implementation available at https://python-visualisation.midas.ijs.si/world-map/sdg-16/

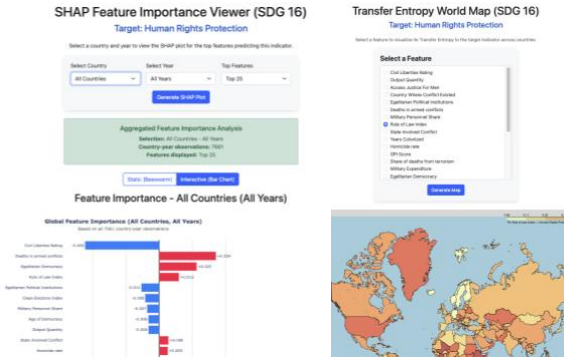


Figure 2: Screenshot of the implementation of SHAP to identify relations between SDG indicators (left), and of the implementation of Transfer Entropy to explore relations of dependency of SDG indicators at country level (right).

A third perspective is offered by StreamStory [4], which abstracts multivariate time series into discrete states and represents their evolution as transitions within a Markov model (see the SDG-focused Streamstory in Figure 3 and the data exploration tool in Figure 4). Rather than examining individual indicator relationships, this approach captures recurring configurations of multiple indicators and the transitions between them. Such a representation can provide a system-level view that complements feature-level predictive importance and pairwise directional information. In particular, relationships identified through SHAP or Transfer Entropy can be examined in the context of broader transitions between developmental states. The previous study proposed this comparison as a way of examining whether strong information-transfer relationships correspond to observable changes in multivariate system configurations.

While SHAP tells us about model-dependent predictive contribution, whereas TE investigates conditional information transfer through time. StreamStory then operates at a higher level, modelling transitions between multivariate states. This provides us with a preliminary technically comparison of three different representations of dependency. These approaches suggest that causal analysis of SDG indicators should not be reduced to the application of a single causal-discovery algorithm. SHAP addresses predictive relevance, Transfer Entropy addresses directional information flow, and StreamStory provides a representation of system-level temporal dynamics. Their outputs therefore describe different aspects of the same underlying process. The central methodological challenge is consequently not only whether an individual method can produce a causal relationship, but whether independently derived representations provide consistent evidence for that relationship.

This issue becomes particularly important when empirical causal structures are compared with causal structures derived from an explicit model of the system. A model encodes assumptions about how variables interact and provides a representation of expected mechanisms, while empirical time-series analysis provides evidence about patterns actually present in observations. Agreement between the two can provide converging evidence for a relationship. Conversely, disagreement can be scientifically informative. A relationship detected consistently in empirical data but absent from the model may indicate an incomplete or incorrectly specified mechanism, while a model-predicted relationship not supported by the data may indicate that the mechanism is weak, masked by noise or confounding, operating at another temporal scale, or not represented adequately in the observations. Thus, comparison between data-derived and model-derived causal structures can be treated not only as a validation exercise but also as a mechanism for identifying potentially new system dynamics.

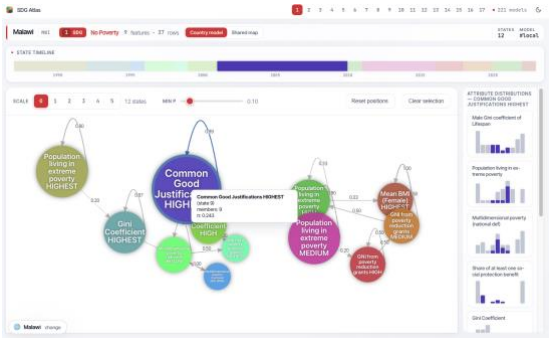


Figure 3: Screenshot of the implementation of Markov chains through Streamstory for SDG analysis

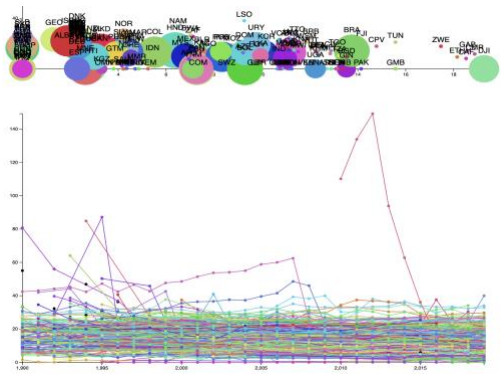


Figure 4: Screenshot of multilevel analysis of time-series data through the SDG Observatory

3 Discussion of Main Results

The results demonstrate that no single analytical approach is sufficient to characterize the dynamics of sparse SDG time-series data. The three methods provide complementary perspectives on the same system. SHAP identifies indicators that are most influential for predicting the target variables, while Transfer Entropy provides information about the direction and strength of temporal dependencies. StreamStory complements these feature-level analyses by representing the evolution of the multivariate system through states and transitions. An important result is the partial divergence between predictive importance and directional information transfer. Indicators with high SHAP importance do not necessarily exhibit strong Transfer Entropy, while some indicators with moderate predictive importance show stronger temporal information transfer. This suggests that predictive

[4] SDG focused instance with implementation available at https://sirius.ijs.si/sdg/

relevance should not be interpreted directly as causal relevance and supports the use of multiple complementary methods when analysing SDG dynamics.

Transfer Entropy (TE) quantifies directional temporal dependence and is applied to multiple SDG indicators and evaluated in both directions, $X \rightarrow Y$ and $Y \rightarrow X$, allowing potential drivers, consequences, and feedback relationships to be identified.

For the SDG analysis, TE is generalized from a simple two-variable relationship into a framework for exploring relationships within a sparse multivariate system. Rather than proposing a new TE formulation, the study extends its application to a sparse multivariate SDG setting, combining TE with SHAP-based predictive relevance and StreamStory-based state transitions to provide complementary evidence for the interpretation and validation of potential causal relationships. The Transfer Entropy results further indicate that relationships between SDG indicators can be directional and potentially bidirectional, with information flowing across different SDG domains. However, given the short and incomplete nature of the available time series, these relationships should be interpreted as evidence of potential causal influence rather than definitive causal effects.

Table 1: Comparative interpretation of the three analytical approaches and causal models

Analytical approach	Primary objective	What it can reveal
SHAP	Identify influential indicators for a target	Which indicators contribute most to prediction of the target
Transfer Entropy	Identify directional temporal dependencies	Potential drivers, consequences, feedbacks, nonlinear and lagged relationships
StreamStory	Characterize system-level temporal dynamics	Recurring development states, transitions, anomalies and broader temporal patterns

Table 1 highlights that the three analytical approaches should be understood as complementary rather than interchangeable sources of causal evidence. SHAP provides an interpretable assessment of predictive relevance, Transfer Entropy extends this analysis by examining directional information flow, and StreamStory represents the data as multivariate states and transitions, allowing recurring developmental configurations and changes in system behaviour to be identified.

Overall, the results support the proposed comparative framework: agreement between complementary methods provides stronger evidence for a relationship, while disagreement identifies relationships that require further investigation. In particular, discrepancies between data-derived relationships and model-derived structures can be treated as potentially informative signals of incomplete model assumptions or previously unrecognized system dynamics. The findings therefore suggest that, for sparse SDG time series, causal discovery is most appropriately viewed as a process of triangulation, validation, and hypothesis generation, rather than as the direct establishment of causal effects.

4 Conclusions & Future Work

This study explored the use of three complementary approaches—SHAP, Transfer Entropy, and StreamStory—for analysing sparse multivariate time series of Sustainable Development Goal indicators. The results illustrate that these methods capture different dimensions of system behaviour: SHAP identifies predictive relevance, Transfer

An important conclusion is that causal discovery in SDG systems should not be considered as the output of a single algorithm. Instead, the combination and comparison of different analytical perspectives can provide a stronger basis for identifying robust relationships and distinguishing potentially meaningful dynamics from relationships that may result from statistical or methodological artefacts. In particular, agreement between approaches can provide additional support for an inferred relationship, while discrepancies can identify relationships requiring further investigation. Future research will therefore focus on two main directions, case-study-based causal discovery, and improving causal methods for specific SDG systems.

Acknowledgments / Zahvala

This research was partially funded by the European Commission's Horizon research and innovation program under grant agreement 10080425 (ELIA).

References / Literatura

[1] J. Renero, R. Maestre, and I. Ochoa, "R e x: Causal discovery based on machine learning and explainability techniques," Pattern Recognition, 2025.

[2] Luka Urbanč, Marko Grobelnik and Joao Pita Costa (2024) Predicting poverty using regression. Proceedings of the SIKDD.

[3] Marko Grobelnik and Joao Pita Costa (2024) An AI-driven Observatory Against Poverty. The Future of Life Institute.

[4] L. Stopar, P. Skraba, M. Grobelnik, and D. Mladenic, "Streamstory: Exploring multivariate time series on multiple scales," IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 4, pp. 1788–1802, 2019.

[5] Our World in Data team, "SDG tracker: Measuring progress to- wards the sustainable development goals," Our World in Data, 2023, https://ourworldindata.org/sdgs.

[6] World Bank, "SDG data bank," (2026) available at: https://databank.worldbank.org/source/sustainable-development-goals-(sdgs), 2025, accessed: 2026-01-15.

[7] W. Zhou, S. Yu, and B. Chen, "Causality detection with matrix-based transfer entropy," Information Sciences, vol. 613, pp. 357–375, 2022.

[8] T. Schreiber, "Measuring information transfer," Physical Review Letters, vol. 85, no. 2, pp. 461–464, 2000.

Pearl, J. Causality. Cambridge University Press, 2009.

Vowels, M. J., Camgoz, N. C., & Bowden, R. (2022). D'ya like DAGs? A survey on structure learning and causal discovery. ACM Computing Surveys, 55(4), 1–36.

Zhang, Z., Zhou, Y., & Zhang, K. (2025). Causal Discovery from Temporal Data: An Overview and New Frontiers. ACM Transactions on Knowledge Discovery from Data, 19(3), 1–28.

Legal Categories from Incident Reports: Evaluating 14 LLMs under the EU AI Act

Stefan Kofilovski
stefan.kofilovski@gmail.com
Ss. Cyril and Methodius University
Skopje, North Macedonia

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

An incident report may describe harm without giving enough facts to classify an AI system under the EU AI Act. We compared 14 large language models with a legal expert's assessment of 144 incident records. A step-by-step legal checklist increased mean exact agreement from 73.2% to 79.5%, a gain of 6.35 percentage points (95% bootstrap interval 2.93–9.87). Macro-F1, which weights categories equally, was almost unchanged. In a separate 784-call comparison on 14 cases, new expanded records barely changed agreement under the basic prompt and reduced it under the checklist prompt. The benefit concerns exact agreement with this reference. The reference reflects the first author's legal assessment under the published rubric; broader claims of legal accuracy require independent validation. Whether models classify by perceived harm remains untested.

Keywords

EU AI Act, legal classification, large language models, AI incidents

1 Introduction

The AI Incident Database (AIID) collects reports of actual or alleged AI-related harm [8]. An AI Act label attached to a report may be read as a legal assessment. Yet the report may omit the system's purpose or conditions of use, both of which can change its category.

In June 2026, the MIT AI Risk Initiative audited ten incidents and found negative quadratic weighted kappa (QWK) for four of eight models on AI Act categories [9]. Its human–human score of 0.56 concerns a separate sample and is not a performance threshold for ours. Earlier work reports uncertainty under draft versions of the Act [1] and gaps in incident documentation [10].

We ask whether a legal checklist improves agreement with an expert and where disagreements remain. We provide the rubric and labels, compare 14 models under two prompts, and examine category and rule-level differences. A separate follow-up compares short and expanded records (Figure 1).

2 Legal Framework and Study Scope

We interpret the results against the July 2024 Act [6]: 113 numbered articles, 13 chapters and 13 annexes, covering risk categories, operator duties, governance and enforcement (Supplement S4). Our rubric asks:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.109

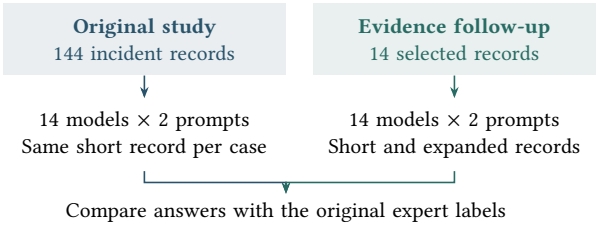


Figure 1: The two comparisons. The expert consulted extra sources for 14 original cases. The follow-up compares model answers under two evidence conditions against the unchanged expert reference.

(1) **Is it an AI system?** Art. 3(1) requires an inferential system. Deterministic logging or human analysis alone does not establish this.
(2) **Is the practice prohibited?** Art. 5 lists prohibited practices, each with legal elements to establish. Art. 5(1)(a), for example, concerns manipulative or deceptive techniques, impairment of informed decisions and significant harm or its likelihood.
(3) **Is the system high-risk?** Art. 6 provides two routes. The product route requires a relevant Annex I product or safety-component role and third-party conformity assessment. The other route covers the listed Annex III uses, subject to Art. 6(3). That exception cannot apply to profiling.
(4) **Do transparency duties apply?** Art. 50 sets transparency obligations for certain uses, including direct interaction and synthetic content. These duties can also apply to high-risk systems.

The rubric prioritises one summary label: prohibited, high, transparency or minimal. These are not four mutually exclusive harm levels in the Act. We record Art. 50 separately and allow not-AI and insufficient-information answers. "Minimal" means no higher rubric category, not harmless or free of legal duties.

The rubric and checklist ask about hypothetical EU placement or use, setting aside territorial and temporal applicability. The basic prompt does not repeat that assumption. Labels are not findings of liability for the reported incidents.

The rubric and checklist also say "today", without fixing a legal version. July 2026 amendments added intimate-material prohibitions, product provisions and an annex [7]. Our historical legal framework is an interpretive scope, not a controlled legal-version experiment. Supplement S13 screens potentially affected records; it does not establish current-law accuracy.

Harm alone does not determine the category, although it matters to tests in Arts. 5 and 6(3). Nor does using a general-purpose tool automatically exempt an actor from Art. 5. Commission guidance covers both deployers and general-purpose systems [5].

Some recorded rubric decisions rely on a narrower view of such misuse; Section 5 explains why those judgments need further review.

3 Data and Methods

3.1 Records and Expert Labels

We sampled 150 of the 1,618 records with titles and descriptions in the 10 August 2026 AIID export (seed 2026). Keywords in titles, descriptions and party fields assigned records to eight Annex III-related buckets or a control pool, resolving ties in a fixed domain order. We drew up to 13 per domain and filled remaining places from the control pool. Buckets encouraged variety without establishing legal labels or coverage.

Six development examples were excluded: S019, S026, S028, S041, S059 and S067. Their notes concern employment, recommendation/moderation, migration, benefit decisions and electoral-purpose boundaries. Selection was not documented as random; the examples do not cover all prohibitions or validate the product route. The original 44 `dev`/106 `heldout` split becomes 42/102. This pooled evaluation includes development records and later rubric revisions.

The 144 evaluation records span 2012–2026. Descriptions have a median of 358 characters (range 64–599). Models received the year, title, description, developer, deployer and harmed parties, without the sampling bucket or web access.

The first author, a lawyer, developed the rubric and labelled the records. These are his expert assessments of the legally correct categories under the stated interpretation and available evidence. Records include six rule decisions, a category, confidence and notes. The correction log lists 19 field changes across ten incidents. The rubric therefore changed during annotation. The author confirms that these revisions preceded his inspection of model answers. No second expert reviewed the labels; inter-annotator agreement was not measured [2].

The final labels are 77 transparency, 44 high, ten prohibited, nine minimal, two not-AI and two insufficient-information. For 14 records, the expert also consulted full pages or linked reports. We report both the full sample and the remaining 130 records to examine this evidence mismatch. Nine of the excluded records are high-risk, so exclusion also changes the category mix.

3.2 Prompts and Measures

Each model received a basic prompt and a longer checklist requesting intermediate rule decisions. Both required a category and short explanation. Legal instructions, scope framing, abstention rules, length and answer structure differed. We compare complete prompts, without isolating a reasoning mechanism.

The checklist's Art. 5(1)(b) summary omits the behaviour-distortion and significant-harm thresholds. Other interpretive assumptions are audited in Supplement S15. Agreement with the archived instruction does not validate its legal completeness.

We use one retained output per model, record and prompt from the original OpenRouter calls: default sampling, a requested 4,096-token completion limit, and no tools. Of 4,032 intended answers, 4,030 were retained; Gemini 3 Flash refused two basic-prompt cases. Original materials are in the public repository; model identifiers, usage-log dates, prompts and input hashes accompany the revision.

Exact agreement is the share of answers matching the expert's label. Macro-F1 gives equal weight to each of the six categories, combining precision and recall within each. Missing answers count as nonmatches and false negatives; undefined category F1 scores count as zero. We also report class-wise scores and confusion matrices. Always answering transparency would give 53.5% exact agreement and 0.116 macro-F1.

QWK [3] accounts for label distributions using the assumed order minimal < transparency < high < prohibited. This is not a statutory harm scale. Missing and auxiliary answers have no ordinal score. Table 1 gives scorable counts; paired changes use each model's records scorable under both prompts.

We resample incidents 10,000 times (seed 20260916), keeping both prompts and all models together, to obtain percentile 95% intervals [4]. These hold labels and outputs fixed, excluding uncertainty in legal judgments, repeated calls and a wider model population. Individual model comparisons are descriptive, without multiple-testing claims.

3.3 Additional-Evidence Comparison

On 22 September, we reran the 14 additional-source cases under both prompts, using fresh short-record baselines and source-linked expansions. Each model received the same record per condition. Thirteen canonical model identifiers match the archive; Qwen3.8 Max uses a disclosed September replacement. Each model used a fixed provider endpoint, with fallback disabled. Original prompts and defaults were retained, although the requested 4,096-token limit did not uniformly cap billed reasoning.

We kept one attempt per cell, including failures, separate from the original results. The original expert labels were retained as a fixed reference and were not reassessed against the new packets. These packets were assembled for the follow-up and do not reconstruct the original annotation evidence. The comparison therefore measures changes in agreement with an existing legal reference. Intervals resample the 14 incidents across models and conditions (10,000 draws, seed 20260922). Repeated-call variability remains unmeasured.

4 Results and Disagreement Analysis

4.1 What Changed with the Checklist?

Mean exact agreement rose from 73.2% to 79.5%, a paired gain of 6.35 percentage points [2.93, 9.87]. Mean macro-F1 barely changed: 0.606 to 0.604, with a paired change of −0.002 [−0.079, 0.081]. Exact agreement increased for nine models and fell for five; the median gain was 4.51 points.

QWK rose for nine models and fell for five (Table 1). The means were 0.424 and 0.498, but the paired change on common scorable records was 0.068 [−0.007, 0.143]. That interval includes zero. Negative QWK means agreement below the baseline implied by the label distributions; it does not mean every random classifier would do better.

Excluding the 14 additional-source cases gave exact agreement of 71.7% and 79.5%. Mean QWK was 0.379 and 0.469, with a paired change of 0.083 [0.001, 0.166]. This checks sensitivity to the affected cases but cannot undo their original evidence mismatch. Excluding 15 automated-driving cases instead left 129 records and a paired QWK change of 0.059 [−0.020, 0.137]. This addresses one recurring case family, not all possible dependence between reports.

On the 102 records marked held-out in the original split, exact agreement increased by 5.32 points [1.47, 9.31]. This post-hoc partition check does not establish an untouched test set (Supplement S13).

Table 1: Agreement with the expert legal reference on 144 incidents. B: basic; S: step-by-step. Exact agreement (%) and six-label macro-F1 include all records. n_o: scorable ordinal pairs for QWK; brackets show 95% incident-bootstrap intervals.

	Exact B/S	F1 B/S	n_o B/S	QWK B [95% CI]	QWK S [95% CI]
Gemini 3.7 Flash	86.8/89.6	0.77/0.69	140/140	0.65 [0.46, 0.81]	0.70 [0.54, 0.85]
Gemini 3 Flash	82.6/83.3	0.54/0.51	138/140	0.57 [0.38, 0.74]	0.53 [0.32, 0.73]
Gemma 3 27B	68.8/66.7	0.29/0.33	140/140	0.28 [0.13, 0.45]	0.31 [0.13, 0.50]
GPT-5.2	81.2/77.8	0.69/0.57	139/140	0.57 [0.39, 0.74]	0.48 [0.30, 0.64]
GPT-5.6 Luna	81.2/72.2	0.73/0.53	137/139	0.56 [0.39, 0.72]	0.30 [0.12, 0.46]
GPT-5.6 Sol	79.2/72.9	0.76/0.75	139/140	0.48 [0.30, 0.65]	0.37 [0.20, 0.53]
Grok 4.6	79.2/87.5	0.74/0.61	139/140	0.47 [0.29, 0.64]	0.67 [0.51, 0.82]
Claude Haiku 4.5	38.9/75.7	0.37/0.52	134/138	-0.07 [-0.15, 0.01]	0.49 [0.31, 0.66]
Kimi K2.5	75.7/87.5	0.68/0.66	139/140	0.45 [0.27, 0.61]	0.65 [0.46, 0.81]
Kimi K3	76.4/74.3	0.53/0.58	139/139	0.54 [0.38, 0.69]	0.42 [0.25, 0.59]
Qwen 3.8 27B	77.8/85.4	0.65/0.79	138/140	0.47 [0.30, 0.64]	0.58 [0.40, 0.74]
Qwen 3.8 Max	79.2/85.4	0.78/0.79	140/140	0.39 [0.20, 0.57]	0.58 [0.41, 0.75]
Claude Sonnet 4.6	54.2/76.4	0.38/0.51	137/140	0.24 [0.12, 0.37]	0.40 [0.21, 0.58]
Claude Sonnet 5	63.2/78.5	0.58/0.62	137/140	0.32 [0.17, 0.47]	0.48 [0.31, 0.65]

4.2 Did More Evidence Help?

We attempted all 784 follow-up calls. Of these, 750 returned valid category answers for the requested incident and 34 did not. Expanded records changed 18 of 187 basic-prompt labels and 26 of 179 step-by-step labels among pairs with valid answers in both conditions.

The unweighted mean of model-specific changes on common valid pairs was 0.18 percentage points [−3.53, 3.85] under the basic prompt and −8.95 [−15.48, −3.23] under the checklist. Counting failed answers as nonmatches gave +0.51 and −7.14 points (Supplement S11). New expanded packets did not consistently increase agreement; the scores cannot determine whether changed answers are legally better.

4.3 Where Did Answers Differ?

Figure 2 separates 109 records labelled outside Annex III and 33 inside; two unclear cases are excluded from this contrast. Outside Annex III, mean signed disagreement fell from +0.24 to +0.12. Inside, it shifted from −0.02 to −0.12. The signs show direction relative to the expert under the imposed category order. Category mix and the ends of that scale matter: a prohibited reference cannot be over-classified, nor a minimal one under-classified. Supplement S6 also compares answers within each reference category.

The most common basic-prompt confusion was transparency to prohibited: 200 of 1,078 answers on transparency cases, falling to 126 with the checklist. In the other direction, prohibited to transparency rose from 48 to 62 of 140 answers. This helps explain why exact agreement and macro-F1 give different pictures. Domain comparisons are limited by very small counts: education and justice/democracy have one reference incident each, employment has two, and critical infrastructure has none.

Checklist answers matched the expert on Art. 5 in 1,544/2,016 model–incident pairs (76.6%). Agreement was 1,921 (95.3%) for the product question, 1,716 (85.1%) for Annex III domain, 1,804 (89.5%) for derogation, 1,705 (84.6%) for profiling and 1,874 (93.0%) for transparency. These rates include negatives and not-applicable values; missing steps count as nonmatches. On Art. 5-positive references, agreement was 51/140 (36.4%; Supplement S14). A first recorded difference does not identify its cause; the basic explanations leave much reasoning unobserved.

Three cases illustrate these differences. S058 describes a fatal incident involving a stacking robot. The expert assigned minimal based on an assumption about product assessment. Twelve basic answers said high and two said not-AI; ten checklist answers said minimal. For S011, facial-image search, ten models under each prompt said prohibited. The expert recorded high because his assessment did not establish untargeted scraping. For S024, activity logs in a cheating investigation, agreement with not-AI fell from eight models to six. The case dossiers include all model labels, contrary explanations and unresolved facts; they are not new human evaluations of the answers.

5 Discussion and Limitations

Disagreement can reflect unsupported assumptions, competing legal readings or a reference judgment needing revision. In S083, the expert's note describes deception, age vulnerability and financial harm, while the corrected label is transparency. Each legal element needs assessment; general-purpose use is not a blanket exemption. Supplement S13 also checks sensitivity to 22 fraud-related transparency records.

We tested how much earlier label changes affect the results by restoring eight documented prohibited-to-transparency corrections to their previous categories. Mean QWK became 0.393/0.437, with a paired change of 0.036 [−0.034, 0.107]. This tests sensitivity to recorded history without changing the original data or validating either legal interpretation.

Neither the reference labels nor the error interpretations have been independently replicated. The audit followed the model runs. AI helped select and prepare cases; the author made the three recorded legal judgments, retaining S014, S024 and S140. Post-run AI suggestions did not change the expert reference. This is guided case review, not human rating of every model explanation (Supplement S12). Development records and rubric revisions limit independence. The sample does not represent incident prevalence; rare classes make macro-F1 unstable, and short reports omit relevant facts. We did not independently measure incident severity. Category composition could explain the directional pattern. The historical framework limits current-law conclusions.

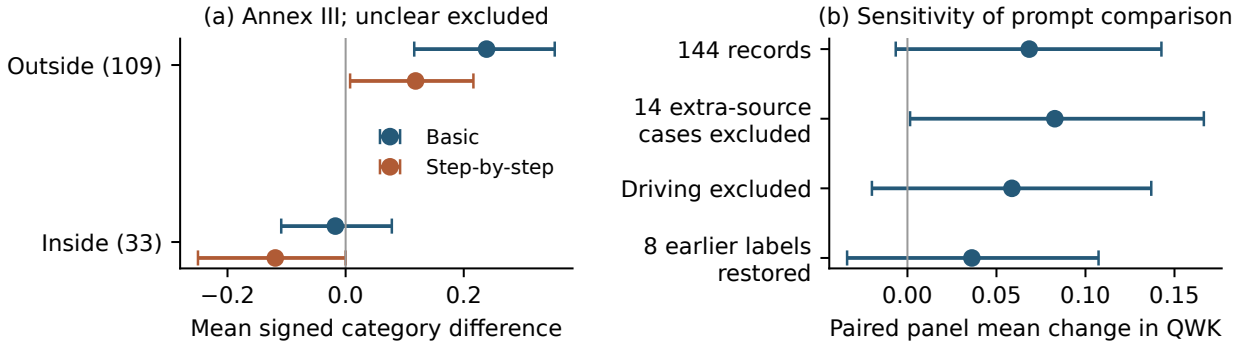


Figure 2: Results with 95% incident-bootstrap intervals. (a) Signed category disagreement by the expert's Annex III classification; two unclear cases are excluded. (b) Checklist minus basic QWK on common scorable records. The eight-label check restores documented earlier labels; neither version has independent legal validation.

6 Conclusion

A legal checklist increased average exact agreement with the expert, but the benefit varied by model and measure. Additional source evidence did not consistently improve agreement. The practical lesson is to report the evidence, rubric and disputed judgments alongside the scores. Independent legal review is still needed before extending these findings to broader claims of legal accuracy.

Acknowledgements

The research was supported through the European project Raido (GA 101135800). LLM assistance was used for manuscript editing, analysis code, source-packet preparation, and selecting and preparing post-run review cases. The authors are responsible for checking the calculations, legal interpretations and final text.

References

[1] appliedAI Initiative. 2023. AI Act: Risk Classification of AI Systems from a Practical Perspective. Tech. rep. https://www.appliedai.de/uploads/files/AI-Act-Risk-Classification-Study-EN.pdf.

[2] Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34, 4, 555–596. doi:10.1162/coli.07-034-R2.

[3] Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70, 4, 213–220. doi:10.1037/h0026256.

[4] Rotem Dror et al. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proc. ACL*, 1383–1392. doi:10.18653/v1/P18-1128.

[5] European Commission. 2025. Guidelines on prohibited artificial intelligence practices established by regulation (EU) 2024/1689. C(2025) 5052 final, 29 July; Section 2.7, paragraphs 39–40. https://digital-strategy.ec.europa.eu/en/library/commission-publishes-guidelines-prohibited-artificial-intelligence-ai-practices-defined-ai-act.

[6] European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). OJ L, 2024/1689. (2024). https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[7] European Parliament and Council of the European Union. 2026. Regulation (EU) 2026/1744 amending regulations (EU) 2024/1689, (EU) 2023/1230 and (EU) 2018/1139. (2026). https://eur-lex.europa.eu/eli/reg/2026/1744/oj.

[8] Sean McGregor. 2021. Preventing repeated real world AI failures by cataloging incidents: the AI incident database. In *Proc. AAAI Conf. on Artificial Intelligence*. Vol. 35, 15458–15463. doi:10.1609/aaai.v35i17.17817.

[9] MIT AI Risk Initiative. 2026. AI incident tracker: June 2026 update. https://airisk.mit.edu/blog/ai-incident-tracker-june-2026-update.

[10] Kevin Paeth et al. 2025. Lessons for editors of AI incidents from the AI incident database. In *Proc. AAAI Conf. on Artificial Intelligence*. Vol. 39, 28946–28953. doi:10.1609/aaai.v39i28.35163.

Balancing Sustainable Development Goals through Orchestrating Autonomous Multi-agent Architecture

Sivaramakrishnan R Guruvayur†
Aquarius & Center for Cybersecurity Systems and Networks Amrita Vishwa Vidyapeetham, Amritapuri, India
sivaramakrishnangr@am.amrita.edu

Krishnasree Achuthan
Center for Cybersecurity Systems and Networks Amrita Vishwa Vidyapeetham, Amritapuri, India
krishnashree@am.amrita.edu

Joao Pita Costa
IRCAI
Institut Jozef Stefan
Ljubljana
joao.pitacosta@ijs.si

Abstract / Povzetek

Sustainable Development Goals (SDGs) motivate a coupled control problem in which interventions targeting one goal may generate synergies or trade-offs with other goals and may also interact with economic, institutional and governance constraints. This paper presents a conceptual design for a hierarchical multi-agent architecture intended to support the analysis of such policy trade-offs while preserving human decision authority. The architecture comprises SDG Goal Guardian and Sub-goal Advocate agents, a Scenario Planner, a Trade-off Balancer, a Macro-Economy Agent, a Governance and Fairness Agent, and a Meta-Orchestrator, with a Human Policy Interface connecting the system to a Policy Leadership Council. Knowledge graphs provide contextual and provenance-aware information, while domain-specific simulation or digital-twin models are proposed as forward models for estimating policy impacts. The proposed decision problem is formulated as a constrained multi-objective optimization problem in which candidate policies are evaluated against SDG objectives and macroeconomic, fairness and governance constraints. Rather than presenting an implemented system or empirical performance results, the paper defines an architecture and an evaluation methodology through which the design could subsequently be tested.

Keywords / Ključne besede

Agentic AI, multi-agent orchestration, Sustainable Development Goals, human-in-the-loop, SDG indicators, AI governance.

1. Introduction

The SDG framework established by the United Nations specifies 17 goals and 169 targets with dense causal interdependencies across economic, environmental, and social subsystems. Interlinkage studies show that interventions targeting a single goal—for example, industrial expansion under SDG 8—can simultaneously drive poverty reduction (SDG 1) and water stress or emissions (SDG 6, SDG 13), depending on local context and macroeconomic conditions. On a positive angle, improvements to the response to water disaster risk reduction (SDG 11) in remote areas allied with the proper education of local communities (SDG 4) can improve climate change resilience of populations (SDG 13), their public health due to fast response (SDG 3), and even food security (SDG 2). It often happens that what affects one SDG will also affect others in that balance. Policy coherence analyses from OECD and the European Commission JRC underline that SDG indicators and macro variables (growth, debt, fiscal space) must be treated jointly; isolated optimization is insufficient for coherent strategy [9]. Agentic AI, where LLM-based agents act as goal-directed entities with tools and memory, has been applied to SDG-related modeling by combining agent-based modeling with autonomous reasoning [2,10]. Multi-agent orchestration surveys [20] and reference architectures identify capability-aligned agents, structured communication protocols (MCP, A2A), and centralized or hierarchical orchestration layers [1,8] as key design elements for scalable, governed multi-agent systems [4,14]. AI governance frameworks and enterprise blueprints further insist on human-in-the-loop (HitL) gates, audit trails, and explicit governance agents for high-stakes decisions [11,18].

We build on this work to propose a hierarchical, capability-aligned multi-agent architecture for SDG balancing. SDG Goal Guardians and Sub-goal Advocates act as strong domain advocates [6,7]; a Trade-off Balancer Agent (TBA) and Macro-Economy Agent compute constrained compromises; a Governance & Fairness Agent (GFA) enforces AI ethics and fairness; and a Human Policy Interface Agent (HPIA) supports a Policy Leadership Council (PLC) that retains final authority under a gated Human-in-the-Loop (HITL) override model. A Meta-Orchestrator Agent (MOA) coordinates these components, grounding reasoning in SDG Knowledge Graphs [3], built on Context Graphs [19] towards a conceptual SDG digital twin through the orchestration of resources [12,15]. In that, this paper addresses the following research question:

Q: What hierarchical multi-agent architecture be designed to address competing SDG policy objectives under fairness and governance constraints while preserving human decision?

To address this, the paper makes a conceptual-design threefold contribution: (i) a hierarchical multi-agent architecture for representing SDG-specific objectives, macroeconomic constraints, governance requirements and stakeholder interests; (ii) a unified constrained multi-objective formulation for evaluating candidate policies across these dimensions; and (iii) an

*Article Title Footnote needs to be captured as Title Note
†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, 5–9 October 2026, Ljubljana, Slovenia
© 2025 Copyright held by the owner/author(s).
http://doi.org/10.70314/is.2026.sikdd.121

evaluation blueprint specifying the data, simulation, stakeholder and policy-coherence tests required for future empirical validation. The proposed architecture, compliant to the UNESCO Recommendation on the Ethics of Artificial Intelligence [13], hasn't yet been implemented nor its optimization performance has been empirically demonstrated.

2. Technical Architecture

A. Layered Design

The architecture (described in Figure 1, image GPT generated) follows a layered multi-agent reference design with clear separation of concerns.

1. *Orchestration Layer:* The Orchestration Layer consists of the Meta-Orchestrator Agent (MOA), Scenario Planner Agent (SPA), Trade-off Balancer Agent (TBA), Governance & Fairness Agent (GFA), and Macro-Economy Agent.
2. *Agent Layer:* The Agent Layer consists of SDG Goal Guardians and Sub-goal Advocates, each encapsulating domain objectives and tools.
3. *Knowledge Layer:* The Knowledge Layer consists of the SDG Knowledge Graph (KG), Context Graph (CG), and Macro-Context Graph (MCG).
4. *Simulation Layer:* The Simulation Layer consists of domain-specific digital twins (DTs) for urban, water, energy, and agriculture.
5. *Human Interface Layer:* The Human Interface Layer consists of the Human Policy Interface Agent (HPIA) and Policy Leadership Council (PLC), implementing human-in-the-loop (HITL) override.

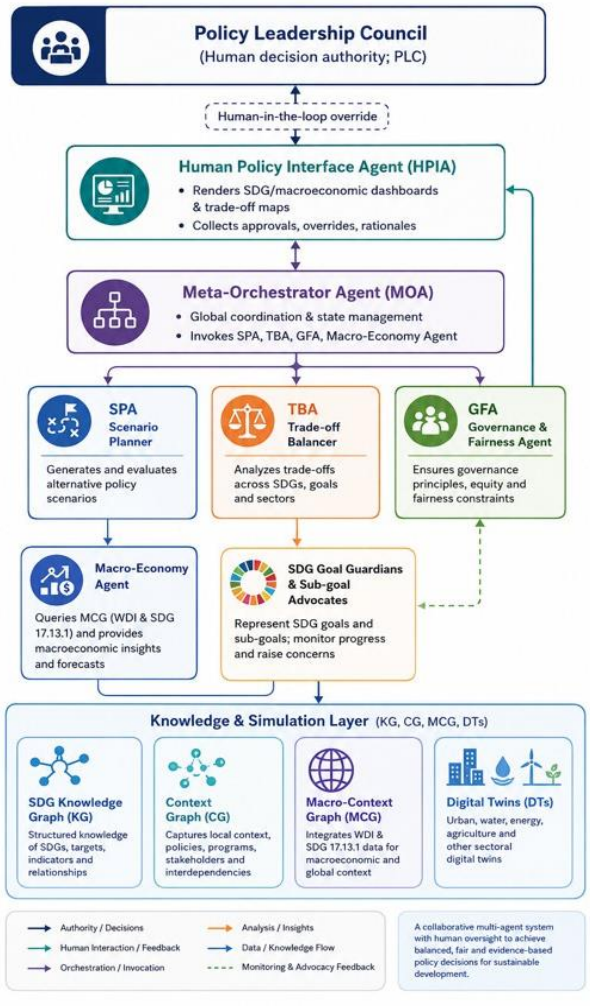


Figure 1: SDG Multi-Agent Orchestration Architecture

B. Knowledge and Macro-Context Representation

The SDG Knowledge Graph (KG) captures goals, targets, indicators, policies, legal norms, and stakeholder entities, built via LLM-assisted extraction and SDG knowledge graph resources. The Context Graph (CG) adds temporal validity, provenance, confidence scores, and local data (e.g., spatial re-source maps and infrastructure), enabling time-aware, context-sensitive reasoning. The Macro-Context Graph (MCG) aggregates World Development Indicators [16] (GDP growth, inflation, debt ratios, employment) and SDG macro-coherence indicators [17] (e.g., 17.13.1), together with policy guidance from the World Bank and IMF resilience and sustainability facilities [5]. A Macro-Economy Agent exposes MCG queries to SPA and TBA, allowing them to impose macro-stability constraints and align scenarios with global economic guidelines.

Digital twins implement forward models for socio-environmental systems; SDG agents query DTs to estimate indicator trajectories and utilities under candidate policies. When f_θ can be mechanistic, data-driven or an ensemble, we get

$$x^t+1=f_\theta(x_t,u_t,z_t)$$

with uncertainty defined as

$$p(x_t+1|x_t u_t)$$

from where we therefore introduce uncertainty-aware policy optimization:

$$\max_\pi \mathbb{E}[J(\pi)] - \lambda \mathrm{Risk}\big(J(\pi)\big)$$

or, more concretely,

$$P\big(U_g(x,u) \geq \tau_g\big) \geq 1 - \alpha_g.$$

C. Agent Roles

Each SDG Goal Guardian Agent is defined by a goal-specific utility function, a set of hard constraints (regulatory and physical limits), and read access to KG, CG, DTs, and selected macro signals. Sub-goal Advocates evaluate specific targets or indicators and feed structured reports to their Goal Guardian. These agents never attempt cross-goal balancing; they act as strong advocates. The Scenario Planner Agent (SPA) decomposes policy questions into a set of scenarios by querying KG/CG/MCG for affected goals and feasibility bounds. The Trade-off Balancer Agent (TBA) aggregates utilities and constraints from SDG and macro agents and computes candidate compromises via multi-objective optimization , introduced as the state transition:

$$x_{t+1} = f_\theta(x_t, u_t, c_t) + \epsilon_t$$

where:

x_t: SDG and macroeconomic state at time t
u_t: policy intervention
c_t: contextual variables
f_θ: learned/simulated transition model
ϵ_t: model uncertainty/noise.

Then define each SDG objective as a trajectory-level utility:

$$J_g(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t U_g(x_t, u_t) \mid \pi\right]$$

where $\pi(u \mid x)$is the policy-selection strategy.

The Governance & Fairness Agent (GFA) applies AI ethics, fairness, and legal constraints, rejecting or flagging non-compliant policies. The Meta-Orchestrator Agent (MOA) manages global state and coordination, while HPIA renders outputs into decision briefs and dashboards for the PLC, which operates under HITL override. Moreover, when x denotes the vector of SDG indicator states and macroeconomic variables, and let u represent policy actions such as investment al-locations, regulatory changes, or infrastructure deployments. The planner's goal is to choose u to optimize a set of objective functions {Ug(x, u)} corresponding to SDG goals g ∈ {1, . . . , 17} and macroeconomic utility Umacro(x, u), subject to constraints C(x, u) capturing feasibility, stability, fairness, and governance. In practice, these objectives and constraints are high-dimensional, non-linear, and context-dependent, making manual planning and isolated models in-sufficient to anticipate emergent behavior and trade-offs.

3. Orchestration Logic & Human-in-the-Loop

A. Trade-off Computation

For scenario s, SDG agents produce a utility vector $\mathbf{U}_{\text{SDG}}(s)$ and constraints $\mathbf{C}_{\text{SDG}}(s)$; the Macro-Economy Agent produces a macro utility $U_{\text{macro}}(s)$ and constraints $\mathbf{C}_{\text{macro}}(s)$. TBA seeks compromise policies $\mathbf{u}^*$ that maximize the combined objectives subject to: (i) SDG floors for each goal; (ii) macro-stability constraints, such as debt, growth, and inflation bounds; (iii) fairness constraints on distributional outcomes; (iv) governance constraints from GFA.TBA approximates Pareto frontiers using evolutionary algorithms, weighted sums, or lexicographic orderings, possibly combined with negotiation among Goal Guardians. It returns a small set of candidate compromise scenarios annotated with utilities and trade-off diagnostics.

B. Orchestrated Decision Cycle

1) Formal Trade-off Balancing: For a given scenario s, let

$$\mathbf{U}(s) = [U_1(s), U_2(s), \ldots, U_{17}(s)] \quad (1)$$

be the vector of goal utilities, and let $U_{\text{macro}}(s)$ denote a composite macro utility derived from MCG, such as growth, debt, and stability indices. TBA's task is to find a policy $\mathbf{u}$ and scenario configuration that solve

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} [U_1(\mathbf{x}, \mathbf{u}), \ldots, U_{17}(\mathbf{x}, \mathbf{u}), U_{\text{macro}}(\mathbf{x}, \mathbf{u})]. \quad (2)$$

subject to:

$$U_g(\mathbf{x}, \mathbf{u}) \geq \tau_g, \qquad g = 1, \ldots, 17, \quad (3)$$

$$C_{\text{macro}}(\mathbf{x}, \mathbf{u}) \leq 0, \quad (4)$$

$$C_{\text{fair}}(\mathbf{x}, \mathbf{u}) \leq 0, \quad (5)$$

$$C_{\text{gov}}(\mathbf{x}, \mathbf{u}) \leq 0. \quad (6)$$

Here, τ_g represents the minimum acceptable floor for SDG goal g. In practice, TBA approximates Pareto frontiers using evolutionary algorithms such as NSGA-II, weighted sums, lexicographic orderings, or market-style negotiation among Goal Guardians, with parameters chosen to reflect policy priorities. TBA produces a small set of candidate compromise policies, each annotated with an SDG utility vector, macro metrics, and trade-off diagnostics, which GFA and HPIA then process.

The orchestrated decision cycle consists of four stages.

1) Problem intake and scenario generation (MOA + SPA): MOA receives a policy question; SPA identifies impacted SDGs and macro dimensions via KG/CG/MCG and generates feasible scenarios.
2) Distributed evaluation (SDG + Macro agents): Goal Guardians and Sub-goal Advocates evaluate scenarios using KG/CG and DTs, computing utilities and constraint violations; Macro-Economy Agent evaluates macro feasibility from MCG.
3) Balancing and governance (TBA + GFA): TBA aggregates reports and computes compromise scenarios subject to SDG, macro, fairness, and governance constraints. GFA checks AI ethics, fairness, and legal compliance and passes compliant scenarios onward.
4) HITL override (HPIA + PLC): HPIA presents candidate policies with SDG trade-off maps, macro dash-boards, and explanations. PLC approves, modifies, or rejects scenarios, with all actions logged and fed back into KG/CG/MCG as institutional memory.

4. Scenario: Water–Energy–Food Nexus

The following illustrative scenario demonstrates how the proposed architecture could address a water–energy–food policy trade-off. It is not an empirical case study and does not use a specific region, observed dataset, calibrated digital twin, or measured outcomes. Consider a semi-arid region where policymakers propose expanding irrigation to increase agricultural production and rural incomes. The Scenario Planner Agent (SPA) identifies SDGs 2, 6, 7, 8 and 13, together with macroeconomic dimensions including agricultural output, rural employment, energy-import dependence and public debt. It then generates alternative scenarios, such as conventional irrigation expansion, drip irrigation with solar-powered pumps, and phased deployment.

The SDG Goal Guardians and Sub-goal Advocates evaluate the scenarios using relevant hydrological, agricultural and emissions models. Conceptually, conventional irrigation could improve agricultural production and rural income while increasing groundwater depletion and emissions, potentially causing SDG 6 and SDG 13 to fall below predefined floors. The Macro-Economy Agent evaluates infrastructure costs, energy-import exposure and debt implications. The Trade-off Balancer Agent (TBA) then identifies a set of feasible, non-dominated alternatives. In this illustrative example, a combination of drip irrigation, solar-powered pumping and phased deployment could provide a feasible compromise across SDG and macroeconomic objectives. These outcomes are hypothetical and are intended only to demonstrate the proposed decision process.

The Governance and Fairness Agent (GFA) evaluates candidate policies against predefined legal, governance and distributional constraints, while the Human Policy Interface Agent (HPIA)

presents their trade-offs to the Policy Leadership Council (PLC). The PLC retains authority to approve, modify or reject the alternatives, with the decision and rationale potentially recorded as institutional memory in the relevant knowledge graphs. The scenario therefore illustrates the proposed sequence of scenario generation, distributed evaluation, multi-objective trade-off analysis and human decision-making without claiming empirical effectiveness.

5. Evaluation Metrics and Limitations

A pilot development of the multi-agent framework integrates domain digital twins, execute case studies, and calibrate optimization algorithms, culminating in evaluation and dissemination. Evaluation metrics span the axes: (i) **Policy Coherence:** Reduction in conflicting SDG im-pacts, quantified via trade-off curves and conflict counts; (ii) **SDG Co-benefits:** Number and magnitude of SDG indi-cators improved per policy decision; (iii) **Equity:** Fairness of outcomes across regions and stake-holder groups, assessed via indices such as Gini and participatory scoring; (iv) **Resilience:** Robustness of recommended policies under simulated shocks and perturbations in digital twin sce-narios; (v) **Explainability:** Clarity and traceability of agent ratio-nales and Orchestrator decisions, evaluated by domain experts; (vi) **Stakeholder Acceptance:** Qualitative feedback from policymakers and practitioners regarding plausibility, utility, and trustworthiness of system recommendations.

Open datasets (SDG indicators, GIS layers, census and environmental data) and standard AI libraries (e.g., PyTorch or TensorFlow for LLM components) form the technical basis, with cloud platforms such as AWS or GCP providing scalable compute for simulation and optimization workloads.

The architecture faces several technical and institutional limitations.

Scalability. As the number of agents, indicators, macro variables, and DTs increases, MOA, SPA, and TBA must handle larger state spaces and message volumes, raising latency and context-window concerns. Hierarchical orchestration, semantic rout-ing, and dynamic agent registries will be needed to maintain scalability.

Data and Uncertainty. KG/CG/MCG quality depends on SDG and macro data that may be incomplete or contested, especially in lower-capacity contexts. Without uncertainty modeling and expert in-put, optimization outcomes may be brittle; robust or Bayesian techniques and continuous data curation are necessary.

Digital Twin Fidelity. DT-based utilities hinge on calibration and maintenance; simplified models can mislead SDG agents and TBA. Validation pipelines, sensitivity analysis, and hybrid human–model oversight must be implemented.

6. Conclusion

The paper defines a common constrained multi-objective formulation through which heterogeneous SDG and macroeconomic objectives can be evaluated, while treating the resulting Pareto set as input to a human policy decision rather than allowing the AI system to determine a final policy autonomously. The architecture also provides explicit locations for representing stakeholder preferences, institutional constraints, uncertainty, provenance and governance requirements.

The water–energy–food example demonstrates the intended orchestration process but is deliberately presented as an illustrative scenario rather than an empirical case study. Consequently, no claim is made regarding the effectiveness, accuracy or policy superiority of the proposed architecture. Establishing these properties requires implementation and systematic evaluation against established SDG decision-support, including system-dynamics and SDG interaction methods.

Future work should therefore focus on a constrained prototype with a limited number of SDGs and stakeholders, validated data sources, explicit provenance, calibrated simulation models and reproducible optimization procedures. Particular attention should be given to stakeholder conflict, uncertainty propagation, LLM reliability, fairness and human decision-making. Such evaluation will determine whether multi-agent orchestration provides measurable advantages over existing approaches for analysing complex SDG policy trade-offs while maintaining appropriate human and institutional control.

Acknowledgments / Zahvala

This research was partially funded by the European Commission's Horizon research and innovation program under grant agreement 101135800 (RAIDO).

References / Literatura

[1] A. Adimulam, et al. The Orchestration of Multi-Agent Systems: Architectures, Protocols, and Enterprise Adoption, arXiv 2601.13671, 2026
[2] I. Chughtai, "AI Agents and Sustainable Development Goals: Bridging the Gap Through Innovation," SBS Journal of App. Business Research, 2026.
[3] E. Fotopoulou et al., "SustainGraph: A Knowledge Graph for Sustainable Development," Front. Environ. Sci., 26, 2022.
[4] R. Kodali, "Governing Multi-Agent AI Systems: An Enterprise Blueprint for Scalable Autonomy," Architecture & Governance, 2026.
[5] IMF, "Resilience and Sustainability Facility—Updated Operational Guidance Note," 2025.
[6] J. Liu et al., "Agentic AI for Sustainable Development: Leveraging LLM-Enhanced Agent-Based Modeling for Complex Policy Strategies", Emerging Media 3.3 (2025): 401-413., 2025.
[7] D. Maheshwari et al., "AI's Global South Pivot: Equity, Ethics and Ecology," CIGI Policy Brief 225, 2026.
[8] Microsoft, "Multi-agent Reference Architecture," 2025.
[9] A. Miola et al, "Interlinkages and Policy Coherence for the Sustainable Development Goals," European Commission JRC 115163, 2019.
[10] OECD, Policy Coherence for Sustainable Development 2019, OECD Publishing, 2019.
[11] SDG Knowledge Hub, "Tools Support Policy Makers to Prioritize Multiple Goals," SDG Knowledge Hub 30, 2020.
[12] T. A. Syed et al., "Agentic AI-enhanced digital twins for Smart City civil infrastructure: A secure, autonomous and auditable management framework" PLOS ONE 21(7), 2026.
[13] UNESCO, "Recommendation on the Ethics of Artificial Intelligence," UNESCO Library, 2021.
[14] J. deVadoss., "Designing Intelligent Enterprise Agents: A Capability-Aligned Multi-Agent Architecture," arXiv 2605.08258, 2026.
[15] L. Wan et al., "From Urban Modelling to City Digital Twins," Environment and Planning B, Urban Analytics & City Science, 53(4), 2024.
[16] World Bank, "World Development Indicators," 2025.
[17] World Bank, "SDG Indicators and Metadata," 2021.
[18] J. Yang et al., "Automating the Construction of Environmental Policy Knowledge Graph with LLMs," Sustainability, vol. 17, no. 22, 10282, 2025.
[19] C. Xu et al. "Context graph." arXiv:2406.11160 (2024).
[20] Y. Zhu et al., "LLM-Based Multi-Agent Orchestration: A Survey of Frameworks, Communication Protocols, and Emerging Patterns," Future Internet, vol. 18, no. 6, 326, 2026.

Accurate but Uninformative: Checking an LLM Assessment Pipeline Without Ground Truth

Martin Macak
Jožef Stefan Institute
Artificial Intelligence Laboratory
Ljubljana, Slovenia
martinmacak@protonmail.com

Marko Grobelnik
Jožef Stefan Institute
Artificial Intelligence Laboratory
Ljubljana, Slovenia
marko.grobelnik@ijs.si

Abstract

Language models are increasingly used to assess systems against a fixed taxonomy: a list of properties, mechanisms or impacts written in advance, from which the model picks those that apply. Such taxonomies are checked for accuracy. That check needs a trusted answer to compare against, and for the downstream effects a system may have there is none. We propose a check that needs none: does the taxonomy describe different systems differently? An element asserted for almost every system carries almost no information about which system it is, and its base rate alone measures how little, as binary entropy, at most one bit per element; summed over a stage's vocabulary this is the stage's capacity. Low capacity proves that a stage cannot distinguish systems; high capacity proves nothing, so the measure is a screen rather than a verdict. Applied to a deployed five-stage AI-risk analyser over 298 system descriptions, 84 of them the AI systems that reach the later stages, it finds the mechanism stage carrying 3.0 of a possible 7 bits. One mechanism is asserted for 81 of the 84 and two are almost never asserted, and no internal consistency check can see either. The same analyser had no scope decision at all. Asking the model directly whether a submission is an AI system scores 0.94–0.97 AUC (area under the receiver operating characteristic curve) across three models; the rule it replaces, which let any submission with one asserted property through, scores 0.51–0.75, or 0.73–0.80 in its strongest form.

Keywords

large language models, structured output, taxonomy design, measurement validity, item analysis, AI system assessment

1 Introduction

Large language models (LLMs) are increasingly used to assess systems against a fixed taxonomy: a list of categories written in advance, from which the model picks the ones that apply to the case in front of it. A *risk taxonomy* names the capability properties a system may have, the mechanisms by which a capability can turn into harm, and the kinds of impact that harm can take. An assessment is the subset of those names the model asserts for one system, with a justification under each. We call each name an *element* of the taxonomy.

The system studied here is a five-stage AI-risk analyser that we built at the Jožef Stefan Institute for policy work (Section 3.1). Its input is a text description of a software system, such as the opening of a repository README. Stage 1 asserts which of 12 capability properties the text supports, Stage 2 which of 7 risk mechanisms follow, Stage 3 which of 18 impact axes are affected, and Stages 4 and 5 write causal chains, analogies and a summary. Each stage is one model call that may refer only to elements the stage above it asserted, and six deterministic checks, the validators, enforce that rule and check that every quoted span is really in the input.

`afero` is a filesystem library for the Go language, a drop-in replacement for the standard `os` package, with no machine learning in it. We submitted the first hundred words of its README. The analyser returned a full assessment: two risk mechanisms, two causal chains, a recommendation that mitigation be coordinated at the organisational level, and two historical analogies, standardised shipping containers and the electrical grid. All four validators the analyser had at the time passed. Nothing in the output is malformed. It is an AI-risk assessment of a filesystem library.

Two things went wrong, and only one of them is visible. The visible one is scope: the analyser had no step that asks whether the submission is an AI system at all, and Section 4.4 adds one. The invisible one is in the vocabulary. The mechanism the analyser asserted for `afero`, `capability_overhang`, is defined in the taxonomy as *"dual-use nature: once deployed, the capability cannot be restricted to its intended use; niche-filling proceeds faster than governance response."* That is true of nearly every deployed capability, so the element is asserted for nearly every system it is applied to. An element that is true of everything is accurate and says nothing, and repairing scope does not repair it.

Taxonomies are checked for accuracy: is each element true of the systems it is applied to? That check needs ground truth, a trusted answer to compare against. For the first stage we have one, because a project's dependency manifest says whether it uses machine learning. For the mechanisms and impacts a system may have, we know of no verified list and no obvious way to build one. In hindsight the question we ask next is obvious; in practice it was not asked, and this analyser was in use without it.

So we ask what can be measured without ground truth: does the pipeline say something *different* for different systems? A stage whose elements are asserted for almost every system, or almost none, cannot distinguish systems. A stage whose elements vary might distinguish systems or might vary at random. Low capacity proves a failure; high capacity proves nothing. We measure capacity over 298 descriptions of software systems (Section 3.3).

Contributions. (1) A *capacity* measure for one stage of a fixed-vocabulary pipeline, computed from element base rates alone, with no labels (Section 3.2). (2) A check of the measure against the one labelled result available, on Stage 1, which shows what it can and cannot do: base rate alone finds the elements to inspect, and only labels say which of them are empty (Section 4.1). (3) The finding: the mechanism stage of this analyser carries 3.0 of a possible 7 bits, with one mechanism asserted for 81 of 84 AI

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.102

systems and two almost never, while the property and impact stages are not dominated (Sections 4.2 and 4.3). (4) A repair of the scope decision, the one decision that does have ground truth: asking the model directly scores 0.94–0.97 AUC against 0.51–0.75 for the property rule it replaced, and 0.73–0.80 for the strongest version of that rule (Section 4.4).

Section 2 places the work, Section 3 describes the analyser, the measure and the corpus, Section 4 reports the measurements, and Sections 5 to 7 say what they show and what follows.

2 Related Work

Work on constrained decoding asks whether a model can be made to emit valid structured output and what that costs in quality [3]: whether the schema is *satisfied*. We ask whether a satisfied schema *says* anything. Hallucination and faithfulness are the usual lens on reliability [4]; here every quote is verbatim and the pipeline still fails.

The closest relative is classical test theory [1], where an item every candidate passes has a discrimination index of zero and is cut. What is new is to treat a policy taxonomy filled in by a language model as a test instrument.

Work on AI risk taxonomies asks about *coverage*: the MIT AI Risk Repository [7] extracts 777 risks from 43 taxonomies, asking what the field misses. Coverage and discrimination are different axes: a taxonomy can cover the space completely and still describe nearly every system the same way.

3 Data and Method

3.1 The analyser

The analyser is a five-stage pipeline. Stage 1 returns the capability properties it finds evidence for, from a fixed list of 12, each with a level from 1 to 5. Levels 2 to 5 assert that the property is present and must carry a verbatim quote from the input; level 1 asserts that it is absent or minimal and carries no quote. Stage 2 asserts a subset of 7 risk mechanisms, the named routes from capability to harm. Stage 3 asserts a subset of 18 impact axes, each with a severity. Stages 4 and 5 write causal chains, historical analogies and two summary assessments. Each stage is one structured model call.

Six validators run over the output. They check that every quoted span is a verbatim substring of the input and that a stage refers only to identifiers the stage above it committed to. Two consequences matter below: a Stage 1 that asserts nothing leaves every later stage nothing to cite, and only the identifiers propagate, since no later stage reads the prose under them.

3.2 The measure

An element *fires* for a system when the stage asserts it. For Stage 1 that means the property appears in the returned list at any level, whether the level says the property is present or absent, because that is what the later stages can cite. Let p be the fraction of systems for which an element fires. The information the element carries about which system this is equals the binary entropy [6] $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$: one bit at $p = 0.5$, 0.29 bits at $p = 0.95$, and zero at $p = 1$. Each stage draws on a fixed list of elements, its **vocabulary**: 12 properties, 7 mechanisms, 18 impact axes. A stage's **capacity** is $\sum H(p)$ over that vocabulary, at most one bit per element. Summing treats the elements as independent, so capacity is a ceiling, and a ceiling in one direction only: elements firing independently at random would make it maximal. Low capacity therefore shows that a stage cannot distinguish systems. High capacity does not show that it can (Section 3.4).

We call an element **uninformative** when $H(p) < 0.30$ bits, the information in a one-in-twenty event, which happens when it fires for fewer than 5.3% or more than 94.7% of systems. The rule we fixed before running screened on base rate instead, at $p \geq 0.67$ or $p \leq 0.05$. That cut charged Stage 1 for two elements that carry 0.81 and 0.86 of a possible bit, so we report the entropy screen and give both counts in Table 2.

A stage is *dominated* if either

(i) uninformative elements make up at least half its vocabulary, or
(ii) its capacity is under half its maximum.

Clause (ii) has one fixed cut, half of the maximum, and nothing else to tune. We wrote the rule, the base-rate screen and a prediction into the analysis script before the 84-system run. A pilot on 22 systems, run through every stage, had already shown `capability_overhang` firing for three quarters of them. The prediction was that Stages 1 and 2 would be dominated and Stage 3 not. Moving the screen to entropy afterwards changes no verdict. Stage 1 was not dominated, and Section 4.1 says what that shows.

3.3 Corpus

The corpus has 298 items in three strata: 106 genuine AI systems, 102 non-AI systems and 90 borderline cases. They come from four sources. Repository READMEs (170) were retrieved programmatically from GitHub topic searches with a star floor, such as `topic:filesystem stars:>500`. Entries from the United States federal AI use-case inventories (95) and court opinions naming a system (13) were retrieved the same way from their published lists. The other 20, such as model cards and reports by non-governmental organisations, were found by hand to fill thin strata, and a hand-found item carries the stratum it was sought for. A README, model card or report item is a verbatim span of about 100 words, cut at whole sentences from the opening of its source; an inventory item is one whole field of the agency's record, and a court item is the passage that describes the system.

For the items found by search, labels were assigned by a rule, not by reading the text. For a repository the rule reads the dependency manifest: "`requirements.txt` declares `torch`" makes an item genuine AI, "no machine-learning dependency in `go.mod`" makes it non-AI. Inventory entries take the agency's own classification, and court cases take how the record characterises the system. The analyser never sees a manifest; it sees only the excerpt.

3.4 What we do not measure

The measure says whether a stage *distinguishes*, not whether it is *right*. A stage can separate systems cleanly and be confidently wrong, and the limiting case is a stage whose every element is a fair coin flip: it reaches maximal capacity and distinguishes nothing. Capacity is a necessary condition, not a sufficient one.

4 Results

Setup. We ran the analyser with three models: `gemini-2.5-flash`, `gemini-3.7-flash` and `gpt-5.6-luna`. Decoding is at temperature 0 with a fixed seed where the model accepts one, and each item was analysed once. Ten calls on one item at these settings returned between five and nine properties, and that variation between draws is not in the intervals below, which resample items. The experiment calls each stage directly, with

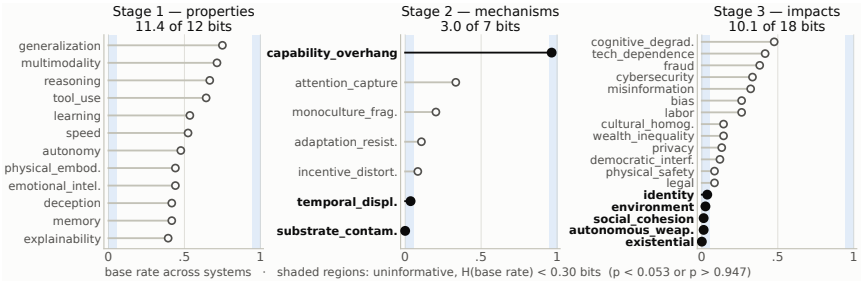


Figure 1: How often each element fires, over the 84 in-scope AI systems. Elements in the shaded regions fire for more than 94.7% or fewer than 5.3% of systems and carry under 0.30 bits each.

Table 1: Checking the screen against a result established with labels. *in-scope*: fraction of the 84 in-scope systems (Section 4) for which Stage 1 asserts the property (`reasoning` is 56 of 84, 0.667 unrounded, under the 0.67 cut); *non-AI*: the same over the 102 non-AI items. *lift* is in-scope minus non-AI, on unrounded rates.

property	in-scope	non-AI	lift
`generalization`	0.75	0.51	+0.24
`multimodality`	0.71	0.42	+0.29
`reasoning`	0.67	0.45	+0.22
`tool_use`	0.64	0.65	0.00
`learning`	0.54	0.40	+0.13

no validator run between stages, so no output was sent back for correction. Stage 1 is measured on all 298 items with all three models. Stages 2 to 5 are measured with `gemini-2.5-flash` only, on the 84 of the 106 genuine AI systems for which Stage 1 asserted at least two properties, which we call the in-scope systems; 10 had none asserted and 12 only one. The study used some 17,000 metered model calls, $41.53 in total.

4.1 Checking the screen against a labelled result

Stage 1 is the one stage with labels, so it is the one place the screen can be checked against a known answer. An earlier labelled analysis had found that `tool_use` does not separate AI from non-AI items on any of the three models. Table 1 gives the five properties Stage 1 asserts most often for the 84 in-scope systems, their rate over the 102 non-AI items, and the difference. `generalization` means the system performs beyond the data it was built on, `multimodality` that it accepts more than one kind of input, `reasoning` that it carries out multi-step inference, `tool_use` that it calls external tools or services, and `learning` that it updates from new data. The analyser decides each from the excerpt alone. Under the same two-property cut on the non-AI side the positive differences shrink to 0.11 or less, so part of the lift is selection.

The screen does not reproduce the labelled finding. It ranks `tool_use` fourth by base rate, at 0.64, below any cut we use, and the two elements the base-rate rule flagged are the two with the largest positive lift. A base rate alone cannot tell a common-and-informative element from a common-and-empty one: `generalization` is common among AI systems and rarer outside them, `tool_use` is common in both, and only the labels separate the two cases. The screen finds the elements worth inspecting; labels, where they exist, say which of them are empty.

Table 2: Capacity in bits by stage, 84 in-scope systems, as defined in Section 3.2. *uninf.*: elements with $H(p) < 0.30$ bits. *p-rule*: elements the base-rate rule fixed before the run would flag ($p \geq 0.67$ or $p \leq 0.05$). (ii) names the clause that fires.

stage	bits	uninf.	*p*-rule	verdict
1 properties	11.4/12	0	2	not dominated
levels 2–5 only	6.5/12	5	5	not dominated
2 mechanisms	**3.0/7**	3	3	**dominated (ii)**
3 impacts	10.1/18	5	5	not dominated

This is also why our prediction for Stage 1 was wrong. We expected it to be dominated because of the `tool_use` finding, and it is not. Telling AI from non-AI is a job the property taxonomy was never designed for and, with no scope step, was pressed into (Section 4.4); telling one AI system from another is the job it was built for.

4.2 Capacity by stage

Table 2 and Figure 1 give the result. `capability_overhang` fires for 81 of 84 systems and carries 0.22 of a possible bit. Two mechanisms fire for three systems or none. The stage carries 3.0 of 7 bits and is dominated under clause (ii). Of the 4.0 bits missing, 0.78 are lost to `capability_overhang`, 1.78 to the two mechanisms that almost never fire, and 1.44 to the remaining four; recovering either of the first two losses alone would flip the verdict. Put plainly, the 84 systems receive only 17 distinct sets of mechanisms, and 37 of them receive `capability_overhang` alone. Stage 1 is not dominated: 11.4 of 12 bits, and no element is uninformative under the entropy screen. Counted on levels 2 to 5 only, the assertions that say a property is present, Stage 1 carries 6.5 of 12 bits with five elements uninformative. It is still not dominated, but two thirds of what Stage 1 commits to the later stages is level 1, an assertion that a property is absent or minimal, which those stages cannot tell apart from a level 5. Stage 3 is not dominated either, at 10.1 of 18 bits, but five of its 18 axes never fire, so its shortfall sits at the opposite end from Stage 2's. That figure counts an impact at any severity. Of the 272 impacts asserted across the 84 systems, 237 are rated low and only 7 are backed by a quote from the excerpt; counted on moderate and above, the stage would carry 1.5 bits.

4.3 Uncertainty in the capacity estimates

A low capacity is not what noise produces, since elements varying at random push it towards the maximum; the plug-in estimate is biased downward by 0.06 bits for Stage 2, which runs against our

verdict, not for it. Bootstrapping the 84 systems with percentile intervals gives 95% intervals of [11.06, 11.53], [2.41, 3.45] and [9.07, 10.74] bits. Stage 2's interval lies below its half-maximum of 3.5 under the percentile method, and reaches [2.54, 3.60] under the bias-corrected method, so clause (ii) is secured by the point estimate, not by the interval, and the element carries the rest of the case: `capability_overhang` sits at 0.96, Wilson 95% [0.90, 0.99]. Over 615 combinations of the two base-rate cuts, 86.7% reproduce all three verdicts and Stage 2's never changes.

4.4 An implicit scope threshold and an explicit scope gate

The analyser had no step that decides whether a submission is in scope: a Stage 1 that asserted no property left nothing to analyse, and one that asserted any property sent the submission through every stage. Whether a system counted as an AI system was therefore a side effect of the citation rule, not a decision anyone made. `afero` was analysed because Stage 1 asserted two properties for it.

We compared two ways of making the decision on the 208 labelled items. The direct question puts the excerpt to the model in one call, with no taxonomy and no property list, and asks whether it describes an AI system on a five-point scale from "definitely not" to "definitely"; the three models, in the order named above, score 0.939, 0.954 and 0.968 AUC, the area under the receiver operating characteristic (ROC) curve. The implicit rule as it stood, any property at any level, scores 0.510, 0.749 and 0.609. The strongest rule we could build from the property count, which counts only properties at levels 2 to 5 and only those selected on held-out folds for their lift, scores 0.765, 0.797 and 0.726, and the paired comparison below is against that stronger rule. The paired bootstrap differences are +0.174 [+0.123, +0.229], +0.157 [+0.107, +0.211] and +0.241 [+0.185, +0.298]. So we added an explicit scope stage before Stage 1. On `gemini-2.5-flash` it cuts the non-AI items analysed from 81 of 102 to 5, at a cost of three more refusals of genuine systems, 13 of 106; the other two models improve on both counts.

5 Discussion

Being true and being useful are different, and only the first gets checked. Every element measured here is defensible: `capability_overhang` really is something deployed capabilities do. An element enters a taxonomy for being true and clearly defined. A third question would have caught the fault: across the systems the element will be applied to, will it ever fire for some and not for others? Answering that costs one pass over a corpus and no labels.

Being consistent is not being right. The six validators check something real: no identifier and no quote is invented. Each compares one stage with the stage above it; none compares any stage with the system that was submitted. The missing check compares a stage's output against the system it describes; the measure here cannot say whether a stage is right, but it can say when a stage could not be.

6 Limitations

The base rates belong to this corpus, mostly README openings taken as they were found; another corpus would move them. The other two models return an empty Stage 1 profile for 124 and 113 of the 298 items, because a README opening says what a project *is* while the taxonomy asks what a system *does*; on five purpose-written descriptions those two models fill in four to five times as many properties, and with `gemini-2.5-flash` the two mechanisms that almost never fire on the corpus each fired once there, in a single run. Stages 2 to 5 are therefore measured on the one model that asserts often enough, and a model that asserts more pushes the common elements towards $p = 1$, so the choice runs towards the dominated verdict, not against it. The 84 systems those stages are measured on are the genuine AI items for which Stage 1 asserted at least two properties, a stricter form of the implicit scope rule Section 4.4 replaces; Stage 1's own base rates are therefore measured on a sample selected by Stage 1, which raises them. A later audit of the excerpts flags 22 of the 84 as changelogs, install notes or link lists rather than descriptions of a system; on the other 62, Stage 2 carries 2.9 of 7 bits and `capability_overhang` fires for 60. Because most labels come from a dependency manifest, any method that reads library names in the excerpt, the direct question included, is partly reading the label. A regular expression over AI vocabulary scores 0.89 on the same 208 items, so the direct question's margin over reading vocabulary is small, although on the 160 of the 208 whose excerpt names no machine-learning library it still scores 0.91–0.95. What we claim is its margin over the property rule in either form. Finally, all three models disagree with the label on 10 of the 208 non-borderline items, 7 of them federal inventory entries; they stay in every number.

7 Conclusion

A taxonomy filled in by a language model can be accurate, internally consistent and still uninformative, and no check internal to the pipeline can tell. Element base rates detect the near-constant elements without labels. In the analyser studied here the mechanism stage carries 3.0 of a possible 7 bits because one mechanism is asserted for 81 of 84 systems and two for almost none, and the scope decision that let a filesystem library in was never made at all. The measure can show that a stage cannot distinguish systems, never that it can. What remains open is checking whether the effects a pipeline predicts are right, the same measurement on other frameworks (OECD [5], the EU AI Act [2], the NIST AI Risk Management Framework [8]), and the direct test: repair `capability_overhang` and measure again.

References

[1] Linda Crocker and James Algina. 1986. *Introduction to Classical and Modern Test Theory*. Holt, Rinehart and Winston, New York.

[2] European Parliament and Council. 2024. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), article 3(1). OJ L, 2024/1689, 12.7.2024, http://data.europa.eu/eli/reg/2024/1689/oj. (2024).

[3] Saibo Geng, Hudson Cooper, Michał Moskal, et al. 2025. JSONSchemaBench: a rigorous benchmark of structured outputs for language models. (2025). arXiv: 2501.10868. DOI: 10.48550/arXiv.2501.10868.

[4] Ziwei Ji et al. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55, 12, 1–38. DOI: 10.1145/3571730.

[5] OECD. 2024. Explanatory Memorandum on the Updated OECD Definition of an AI System. OECD Artificial Intelligence Papers 8. OECD Publishing, Paris. DOI: 10.1787/623da898-en.

[6] Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27, 3, 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[7] Peter Slattery et al. 2024. The AI risk repository: a comprehensive meta-review, database, and taxonomy of risks from artificial intelligence. (2024). arXiv: 2408.12622v1. DOI: 10.48550/arXiv.2408.12622.

[8] Elham Tabassi. 2023. Artificial Intelligence Risk Management Framework (AI RMF 1.0). Tech. rep. NIST AI 100-1. National Institute of Standards and Technology. DOI: 10.6028/NIST.AI.100-1.

Combining Supervised and Unsupervised Learning on SMART Data for Hard Drive Failure Assessment

Luka Petek
University of Maribor, Faculty of Electrical Engineering and Computer Science
Maribor, Slovenia
luka.petek1@student.um.si

Vili Podgorelec
University of Maribor, Faculty of Electrical Engineering and Computer Science
Maribor, Slovenia
vili.podgorelec@um.si

Abstract

Unexpected disk failure is one of the most common causes of downtime in storage systems and data loss. However, the built-in SMART warnings on modern drives catch only a small fraction of actual failures. Manufacturer-embedded SMART thresholds detect only 3–10% of failures at acceptably low false-alarm rates. Machine learning can be used to improve failure assessment. Classical methods (SVM or Naive Bayes) perform reasonably well on SMART data, but most of them rely on a single model and do not combine different learning strategies. Four models spanning supervised and unsupervised learning are combined: a random forest classifier, an unsupervised autoencoder-based anomaly detector, a classifier built on the autoencoder bottleneck, and an HDBSCAN clustering model. Each model captures a different kind of failure signal and has its own interpretational value. All 4 model scores are combined into a single Aggregated Health Index (AHI). On unseen evaluation data from the first quarter of 2026, AHI achieves a ROC-AUC of 0.923 and a PR-AUC of 0.936, and the final AHI can be decomposed into the individual contributions of each model, providing an explanation of the assessment.

Keywords

hard drive failure assessment, SMART data, hybrid model, autoencoder, anomaly detection, HDBSCAN clustering, interpretable machine learning

1 Introduction

Hard disk drives are among the most frequently replaced components in storage systems, and their unexpected failure is a leading cause of data loss, service unavailability, and increased operational cost [1]. To provide early warning, drive manufacturers embed Self-Monitoring, Analysis, and Reporting Technology (SMART) into virtually every modern drive. However, the threshold-based failure flags built into SMART firmware are deliberately conservative: manufacturers report failure detection rates of only 3–10% at an acceptable false-alarm rate of roughly 0.1% per year [2], leaving the majority of failures undetected until it is too late.

Machine learning methods are a viable approach to improving these basic failure signals. Various machine learning methods and models have been recently used to assess hard drive condition based on historical operational data, specifically SMART (Self-Monitoring Analysis and Reporting Technology) attributes [3]. However, each machine learning method has a distinct blind spot. No single model covers all failure modes: gradual degradation may go unnoticed by a supervised classifier trained on binary labels, while explicit failure signals may be missed by an unsupervised anomaly detector that never saw them. Murray et al. [2] established a foundational comparison between SVMs and clustering against manufacturer thresholds but did not integrate the paradigms. Ahmed and Green [4] addressed class imbalance with a cost-aware LSTM but remained within a single supervised paradigm. Xu et al. [5] predicted a continuous health degree with noise-based training in a semi-supervised setting, while Zhou et al. [6] used reconstruction-based anomaly detection to identify deterioration modes – both, however, committed to one learning strategy. This is where a hybrid approach that combines multiple paradigms into a single, interpretable output stands out.

The main contributions of this paper are:

- Four models spanning supervised and unsupervised learning – a random forest classifier, an unsupervised autoencoder anomaly detector, a classifier built on the autoencoder bottleneck, and an HDBSCAN clustering model – are combined into a single Aggregated Health Index (AHI).
- All four per-component scores remain visible alongside AHI, ensuring the final output is interpretable and traceable to each model's contribution.
- Evaluation results and the trade-offs of using this hybrid as a small, fully offline pipeline for data-centre and edge deployment are reported.

The remainder of this paper is organised as follows. Section 2 reviews related work on SMART-based failure prediction. Section 3 describes the dataset, preprocessing, the four component models, and the AHI aggregation formula. Section 4 reports evaluation results. Section 5 discusses deployment use cases, trade-offs, and limitations, while Section 6 concludes the paper.

2 Related Work

Machine learning (ML) applied to raw SMART attribute values, rather than the manufacturer's own threshold flags, has been shown to improve detection rates [2, 1]. Yet ML-based disk failure prediction remains fragmented. Supervised classifiers, such as the SVMs used by Murray et al. [2], achieve high precision when enough labelled failures are available but struggle with the extreme class imbalance typical of SMART datasets (often exceeding 10,000:1 [4]). Unsupervised anomaly detectors, such as autoencoders trained on healthy drives only, do not require failure labels but cannot exploit known failure patterns when they exist. Clustering approaches group drives by operating regime and provide population-level risk context, but do not output a direct failure probability. Semi-supervised methods, such as the reconstruction-based approach of Zhou et al. [6] and the health-degree prediction of Xu et al. [5], attempt to bridge this gap but still rely on a single learning strategy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.97

3 Data and Methodology

Dataset. The study uses Backblaze's 2025 hard drive SMART dataset [7], which covers daily records from the prepared 2024-10-01–2025-09-30 period. The fleet is mostly HDDs [8], and failures are rare compared to healthy records (high class imbalance), which was one of the main challenges.

Two dataset variants. Two training datasets were prepared:

- *Healthy-only sample* (for autoencoder training): up to 1,000 healthy rows per daily CSV were sampled, giving approximately 362,463 healthy rows, split by serial number into 289,982 for training and 72,481 for validation and threshold selection. No failure rows were used.
- *Balanced 50:50 set* (for supervised models and clustering): all 4,414 failure rows plus 4,414 healthy rows sampled at random, giving 8,828 records total. Used for training the random forest and bottleneck classifier, and for learning HDBSCAN clusters.

Prediction target. Each record is a single daily SMART snapshot. Positive examples are failure-day records — the daily report on which a drive was marked as failed. Healthy instances are snapshots from drives that did not fail on that day. The system does not track a drive over multiple days and does not use a defined lead time (e.g. 1 or 7 days before failure). A healthy label on one day therefore does not imply long-term health; it only means the drive had not failed by that date.

Preprocessing. Capacity bytes are converted to gigabytes, an SSD indicator is derived from the model name, missing counters are filled with 0, medians are used for a few mechanical counters, and three engineered features are added: `any_critical_error`, `total_error_count`, and `error_per_gb`. Neural models use MinMax scaling; the scikit-learn model uses the same processed features without the NN scaler.

Features. A set of 19 features is used:

- `capacity_gigabytes`, SSD indicator
- SMART raw counters: 1, 3, 4, 5, 7, 9, 12, 187, 188, 191, 192, 193, 197, 198
- Engineered: `any_critical_error`, `total_error_count`, `error_per_gb`

Models. The full implementation is available at [9]. All four models start from the same 19 processed features but differ in their input representation and training data:

(1) *Random forest* (scikit-learn): input is the 19 features plus manufacturer one-hot encoding. Trained on the balanced 8,828-row dataset, which is grouped by serial numbers, with train/test splitting and GroupKFold cross-validation. Outputs a failure probability in $[0, 1]$.
(2) *Anomaly autoencoder* (TensorFlow): input is the 19 features after MinMax scaling, passed through a symmetric encoder–decoder with a 12-dimensional bottleneck. Trained only on 289,982 healthy rows; 72,481 healthy rows used for validation and threshold selection (99th percentile of reconstruction error). Failure rows were never used for training. Outputs a 0–1 anomaly score.
(3) *Bottleneck classifier* (TensorFlow, two-stage): a separate autoencoder with a symmetric encoder–decoder and an 8-dimensional bottleneck is trained on the same healthy training rows, grouped by serial numbers. Its 8-dimensional encoder is then frozen. A small dense classifier with hidden layers of 16 and 8 units and a single sigmoid output is trained on the 8-dimensional bottleneck features of the balanced 8,828-row dataset, split 70/15/15 (train/val/test). Output is a failure probability in $[0, 1]$.
(4) *HDBSCAN clustering*: input is the same 8-dimensional bottleneck from the classifier autoencoder. Fitted on the serial-grouped training partition; held-out serials are used for internal evaluation. Each cluster is assigned a risk score equal to its failure rate in the training data.

The bottleneck classifier and HDBSCAN share the same 8-dimensional encoder, so their outputs are not independent. The anomaly autoencoder is a separate model with its own 12-dimensional bottleneck.

Aggregation. To obtain a single score, an Aggregated Health Index (AHI) is computed as a weighted root-mean-square of the four component scores, then clipped for stability. Fixed manual weights were used (RF 0.30, bottleneck classifier 0.40, anomaly 0.20, clustering 0.10), and the final AHI can be explained from per-component contributions.

4 Results

As this is a hybrid approach, each of the 4 components is evaluated independently first, and their combined performance is then assessed using the AHI. For a direct comparison, all components and AHI are evaluated on exactly the same strict-unseen records from the first quarter of 2026. The main metrics are summarised in Table 1. The evaluation uses equal numbers of failure and healthy records, so this balance does not represent the real fleet.

Table 1: All components and AHI evaluated on the same strict-unseen first quarter of 2026 records (1,018 failure and 1,018 healthy records).

Model	ROC-AUC	PR-AUC	Recall	FPR	Precision	F1
Random Forest	0.922	0.938	0.812	0.067	0.924	0.865
Bottleneck classifier	0.906	0.912	0.772	0.086	0.899	0.831
Anomaly detector	0.737	0.742	0.522	0.064	0.891	0.658
HDBSCAN cluster risk	0.853	0.816	0.820	0.227	0.783	0.801
AHI	0.923	0.936	0.783	0.091	0.896	0.835

4.1 Random Forest Classifier

A random forest classifier was trained on a balanced 50:50 dataset using serial-number-grouped splitting and GroupKFold cross-validation. On the common evaluation set from the first quarter of 2026, it reaches ROC-AUC 0.922, PR-AUC 0.938, recall 0.812, precision 0.924, and F1 0.865. The model catches most failing drives while keeping false alarms relatively low. The random forest results are solid and it has the highest F1 and precision among the listed components, while AHI has a slightly higher ROC-AUC.

4.2 Autoencoder Anomaly Detector

The anomaly detection model was trained only on healthy rows and evaluated on the common evaluation set from the first quarter of 2026. Reconstruction error reasonably separates failures from healthy records, with a ROC-AUC of 0.737 and PR-AUC of 0.742. However, with a failure recall of 0.522, precision of 0.891, and F1 of 0.658, the model still misses some actual failures. Its main value is that it does not require any failure labels during training, and it catches a different kind of problem than supervised classifiers do, which could potentially contribute to the overall performance of the combined model.

4.3 Bottleneck Classifier

The bottleneck classifier is a strong individual model and consists of two parts. First, the autoencoder compresses the 19 input features into an 8-dimensional bottleneck — a latent data representation — and then a small dense classifier is trained on those 8 dimensions using a balanced 50:50 dataset. On the common evaluation set from the first quarter of 2026, it reaches ROC-AUC 0.906, PR-AUC 0.912, recall 0.772, precision 0.899, and F1 0.831. Its internal grouped test metadata reports ROC-AUC 0.928 and PR-AUC 0.940, but the common first quarter of 2026 result is the primary comparison because every component is evaluated on the same records.

4.4 HDBSCAN Clustering

The clustering model also works on the same 8-dimensional bottleneck data representation from the autoencoder. HDBSCAN [10, 11] is applied directly in that 8-dimensional data space; the number of clusters is determined automatically from the data density. In the grouped training partition, 21 clusters and one outlier group emerged, with 20.7% of points classified as noise. Each cluster is given a risk score equal to its failure rate in the training data; the outlier group had a 66.4% failure rate. When evaluating a new disk, it is placed into a cluster based on the local density structure using HDBSCAN's `approximate_predict` procedure. On the common evaluation set from the first quarter of 2026, cluster risk reaches ROC-AUC 0.853, PR-AUC 0.816, recall 0.820, precision 0.783, and F1 0.801.

4.5 Aggregated Health Index

The Aggregated Health Index (AHI) is simply a weighted root-mean-square (RMS) formula that combines all 4 model scores into one final value representing the current risk state of the disk:

$$\mathrm{AHI} = \sqrt{\frac{w_K \cdot K^2 + w_R \cdot R^2 + w_A \cdot A^2 + w_C \cdot C^2}{\sum w}} \times 100 \quad (1)$$

where K is the random forest failure probability ($w_K = 0.30$), R is the bottleneck classifier failure probability ($w_R = 0.40$), A is the autoencoder anomaly score normalized to $[0, 1]$ ($w_A = 0.20$), and C is the empirical failure rate of the assigned cluster ($w_C = 0.10$). All scores are within the range $[0, 1]$, weights sum to 1, and the result is multiplied by 100 to express AHI on a 0–100 index. The final value is clipped to $[3,\ 97]$ to avoid extreme boundary values. AHI is a risk index, not a calibrated probability: higher values indicate a more failure-like current SMART state, but the RMS aggregation means the output is not calibrated against real-world failure rates. Values below 45 mean *Healthy*, 45–65 is *Warning*, and above 65 is *Critical*.

One property of the RMS formula is that a single strong signal from a low-weight component may not be enough to cross a threshold on its own. For example, if the anomaly score is 1 and all other scores are 0, AHI is $\sqrt{0.20} \times 100 \approx 44.7$, which remains in the Healthy range. This is a design trade-off: RMS amplifies large individual signals, but the weight still scales the contribution, so a single low-weight component cannot dominate the final score by itself.

AHI was evaluated on a strict-unseen evaluation set from the first quarter of 2026 [12] containing 1,018 failure-day records and 1,018 healthy records. Figure 1 shows the AHI scores for both groups.

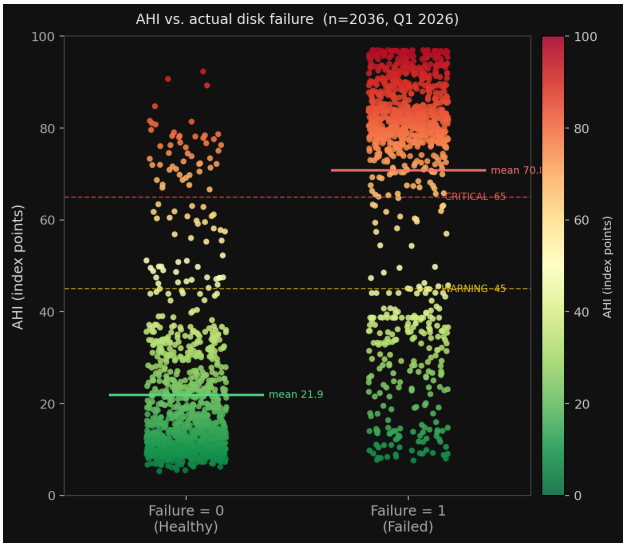


Figure 1: AHI scores for the strict-unseen evaluation from the first quarter of 2026. Each point is one daily SMART record; the vertical axis shows AHI index points.

AHI reaches ROC-AUC 0.923, PR-AUC 0.936, recall 0.783, precision 0.896, and F1 0.835 at the fixed 45-point threshold. Because the evaluation set is balanced by design, its precision must not be interpreted as precision at the natural fleet failure prevalence.

5 Discussion

The results show that no single model is the best at everything. The random forest has the highest precision and F1, while AHI has the highest ROC-AUC and PR-AUC. The autoencoder-based anomaly detector catches a different kind of failure – one it can find even without seeing any failure examples during training. The clustering model adds something different on top of that: each cluster has a human-readable description of its dominant SMART pattern, so both the AHI score and the type of disk behaviour causing it can be examined. The main value of the hybrid approach is not the final AHI alone, but also the ability to look at each of the four component scores individually and critically assess what each model is saying about the drive.

The evaluation is balanced and uses failure-day positives, so it measures current-state rather than future failure prediction. It does not by itself show that AHI is always better than the individual models. The specific causes of false negatives remain unknown. Also, because this evaluation set is balanced, its precision is not the same as in a real fleet where failures are much less common. For example, if failures represent 0.1%, 0.5%, or 1% of records, the estimated precision would be approximately 0.85%, 4.1%, and 8.0%, respectively.

Two further limitations are worth noting. First, all training data comes from a single storage provider (Backblaze), so results may not generalise directly to other hardware vendors or real-life scenarios. Second, the system outputs the current health state of a drive rather than an estimated probability or time of a future failure — remaining life estimation is left for future work.

The implementation also includes a React dashboard that shows AHI, the four component scores, disk information, and analysis history. Setup instructions, a dashboard screenshot, and API usage are available in the GitHub README [9]. The pipeline

works locally and does not need a remote service. A small CPU test on the development computer showed the model files used to evaluate one disk were about 7.1 MB in total. Evaluating one disk took about 0.20 s. The program used about 0.40 GB of memory after loading the models and about 0.95 GB during normal evaluation. Testing on real NAS or edge hardware is left for future work.

6 Conclusion and Future Work

This paper presents and evaluates a hybrid system that combines four models into a single Aggregated Health Index (AHI): a random forest classifier, an autoencoder-based anomaly detector, a bottleneck classifier, and an HDBSCAN clustering model. Each model captures a different kind of failure signal, and the weighted RMS formula combines these signals into one current-state index. On the same 2,036 unseen records from the first quarter of 2026, AHI achieved ROC-AUC 0.923, PR-AUC 0.936, precision 0.896, recall 0.783, and F1 0.835 at the fixed 45-point threshold. These results support current-state assessment, but they do not show how likely a drive is to fail in the future, how early a failure could be detected, or that AHI is always better than each individual model.

For future work, it would be good to tune the weights based on data rather than setting them manually, perform a full SMART feature analysis for each cluster to improve the cluster descriptions, and extend the system to NVMe drives, which have a different set of SMART attributes than traditional HDDs. Evaluation on a naturally imbalanced dataset that reflects real fleet failure rates is also left for future work.

References

[1] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. 2020. Making disk failure predictions smarter! In *Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST)*. USENIX Association, 359–372. https://www.usenix.org/conference/fast20/presentation/lu.

[2] Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado. 2005. Machine learning methods for predicting failures in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, 6, 5, 783–816. https://www.jmlr.org/papers/v6/murray05a.html.

[3] Elizabeth Atekoja. 2024. Prediction of hard drive failure using machine learning. *Global Journal of Computer Science and Technology: D Neural Artificial Intelligence*, 24, 1.

[4] Jishan Ahmed and Robert C. Green II. 2024. Cost aware lstm model for predicting hard disk drive failures based on extremely imbalanced s.m.a.r.t. sensors data. *Engineering Applications of Artificial Intelligence*, 127, Part B, 107339. doi:10.1016/j.engappai.2023.107339.

[5] Chuanmao Xu, Guangyan Wang, Xie Liu, Deke Guo, and Tian-Yi Liu. 2018. Layerwise perturbation-based adversarial training for hard drive health degree prediction. *arXiv preprint arXiv:1809.04188*. https://arxiv.org/abs/1809.04188.

[6] Hao Zhou, Zhiheng Niu, Gang Wang, Xiaoguang Liu, Dongshi Liu, Bingnan Kang, Zheng Hu, and Yong Zhang. 2024. Proactive drive failure prediction for cloud storage system through semi-supervised learning. *IEEE Transactions on Dependable and Secure Computing*, 21, 4, 1528–1543. doi:10.1109/TDSC.2023.3286093.

[7] Backblaze Inc. 2025. Hard drive test data — open dataset. https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data. Daily SMART snapshots from Backblaze's production storage fleet. (2025).

[8] Andy Klein, Stephanie Doyle, and David Johnson. 2026. Backblaze drive stats for 2025. https://www.backblaze.com/blog/backblaze-drive-stats-for-2025/. Annualized failure rates and lifetime statistics for 344,196 drives. (2026).

[9] Luka Petek and Vili Podgorelec. 2026. DiskGuard: hybrid hard drive failure assessment using SMART data. https://github.com/Luka-Petek/DiskGuard. (2026).

[10] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2013)* (Lecture Notes in Computer Science). Vol. 7819. Springer, 160–172. doi:10.1007/978-3-642-37456-2_14.

[11] Leland McInnes, John Healy, and Steve Astels. 2017. Hdbscan: hierarchical density based clustering. *The Journal of Open Source Software*, 2, 11, 205. doi:10.21105/joss.00205.

[12] Backblaze Inc. 2026. Hard drive test data — first quarter 2026. https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data. Daily SMART snapshots from 1 January to 31 March 2026. (2026).

Advertising Is Not Noise: Sponsored Speech Outranks New Terminology in Automatic Term Extraction

Martin Macak
Jožef Stefan Institute
Ljubljana, Slovenia
martinmacak@protonmail.com

Abstract

Long-form podcasts are an attractive corpus for terminology extraction: unedited, precisely dated and speaker-attributed. They are also sponsored, and a sponsor read is a script performed word for word across dozens of episodes, which is exactly what one of the two standard term-extraction signals is built to find. Collocation strength rewards phrases whose words recur together far above chance; rarity rewards words that are unusual in general English. We show on 788 episodes of one science-and-current-affairs podcast (10.2 million words, 2017–2026) that the two signals treat advertising in opposite ways, even after a cleaning step has removed most of it. Of the 100 phrases collocation ranks highest, the 87 that occur in episodes with third-party advertising labels spend a median 100% of their uses inside a marked advertisement, against a base rate of 4.7%; the top 100 by rarity and frequency-matched random phrases sit at 0%. The reason is mechanical: fixed wording raises collocation, while the brand names that would raise rarity are what a cleaner removes first, when it catches them. This is a claim about the head of the ranking, where a pipeline takes its cut, not about whether either signal separates advertising from terminology across their whole distributions; we could not test that cleanly, and we say why. The collocation result repeats on seven of eight other podcasts. We also report the cleaning step's recall and precision, and release a hand-checked lexicon of 59 coined terms.

Keywords

terminology extraction, spoken corpora, corpus contamination, collocation, neologism, podcast mining

1 Introduction

Podcast and video transcripts have become a standard corpus type in computational social science. Long-form spoken corpora are attractive because they are unedited, precisely dated, speaker-attributed, and available at a scale that written interview data never reaches.

They are also sponsored. Advertising is read aloud from a script, repeated close to verbatim across dozens of episodes, and built around brand names that do not occur in general English. Fixedness, repetition and lexical distinctiveness are what automatic term extraction looks for. The standard treatment of advertising in corpus work is to remove it as noise before analysis begins and to validate the removal by a gain on a downstream task. We argue that this understates the problem: removal is never complete, and what survives is not inert.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.103

Automatic term extraction rests on two signals. *Collocation* asks how much more often the words of a phrase occur together than chance would predict; the usual measure is pointwise mutual information (PMI). *Rarity* asks how unusual those words are in everyday English. Both are in routine use, and a given pipeline usually ranks its candidates by one or the other. Combined scores exist [8], but their weighting is a free parameter, so which signal dominates a ranking remains a choice.

Claim. On a sponsored podcast corpus that has already been cleaned, collocation ranks the advertising that survives cleaning above genuine coined terminology, and rarity does not, because the cleaner has already taken the brand names out. The claim is about the top of the ranking, which is where a pipeline takes its term list. The phrase collocation ranks highest in our cleaned corpus is `midnight luxe hybrid mattress won`, a mattress advertisement. We are aware of no prior work that evaluates term-extraction signals against advertising as a labelled competing class on a spoken corpus, or that reports the two signals disagreeing about it.

Any study of terminology, new words or discourse on a sponsored corpus will therefore carry advertising in its results without saying so, and how much depends on which signal it ranked by, a choice usually made on unrelated grounds.

Contributions. (1) The asymmetry, measured: of the 100 phrases ranked highest by collocation, the 87 falling in labelled episodes spend a median 100% of their uses inside a segment a third party marked as advertising, while the top 100 by rarity and two sets of random five-word control phrases sit at 0%, against a 4.7% base rate (Section 4.1). (2) The mechanism: a sponsor read is a fixed expression repeated verbatim, which raises collocation, while the brand tokens that would raise rarity are what a cleaner removes (Section 5). (3) A cleaning step with both error rates: 84.9% recall against 318 third-party segments, 4.9 times a repetition-only baseline, at 47.4% calibrated precision, so about 4 percentage points (pp) of the corpus is genuine content removed with the advertising (Section 4.2). (4) A released lexicon of 59 coined terms in 64 surface rows, each dated to its first use (Section 4.3). (5) Ranking is not separation: whether either signal separates coinages from advertising is decided by how the comparison class is built (Section 4.1). (6) A replication on eight other podcasts (Section 6).

Section 2 reviews term extraction and corpus cleaning, Section 3 the corpus, labels and detector, Section 4 the ranking comparison and the detector's cost, and Sections 5 to 7 the mechanism, its limits and what follows.

2 Related Work

Term and keyphrase extraction. Contrast against a reference corpus (keyness [12], weirdness [1]) assumes a term is vocabulary the general language does not use. Internal cohesion (PMI [4], likelihood-ratio collocation tests [6], C-value/NC-value [8]) assumes its parts co-occur far above chance.

Neologism detection. New-word detection is dominated by *written* monitor corpora: newspapers [7] and large social-media pipelines [16]. Closest is Rossini and van der Plas [16], who reduce a very large token set to a hand-verified shortlist; their corpus is Reddit, written and unsponsored, so this contamination cannot arise there. The marker detector used here follows the pattern-extraction tradition [9].

Podcast and speech-recognition corpora. Large podcast corpora are recent: the Spotify Podcast Dataset [5] and the Text Retrieval Conference (TREC) Podcasts Track [10], built for search and summarization [11].

Corpus cleaning, and where this work departs from it. Removing non-content is a mature subfield: boilerplate detection [13], CleanEval [3], web-corpus construction [2]. In podcasts, Reddy et al. [15] detect the same material as the detector of Section 3.2 and validate it by showing that removal improves summarization. Across all of this work advertising is *noise* to be removed. That framing is right but incomplete: removal is never total, and one of the two standard signals prefers what survives.

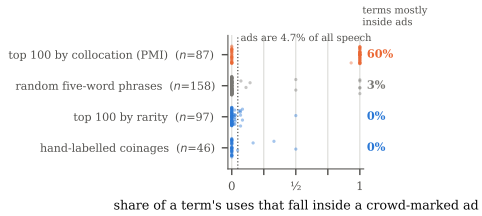


Figure 1: The two standard signals disagree about the same cleaned corpus. Each dot is one term, placed by the share of its uses that fall inside advertising as marked by SponsorBlock's volunteers. Ranked by collocation, in orange, the median term spends every one of its uses inside advertising; ranked by rarity, in blue, none does. Grey is the control: random five-word phrases matched on how often they occur. The right-hand column counts the terms that fall mostly inside advertising. The label comes from outside this project, not from the detector of Section 3.2.

3 Data and Method

3.1 Corpus and labels

The corpus is 788 dated episodes of a single long-running science-and-current-affairs podcast with two regular hosts: 829,989 diarized utterances, 10.23 million words and 1,042 hours of speech, recorded between 10 June 2017 and 13 May 2026. The audio was transcribed with Deepgram's `nova-3` model with speaker diarization.

We measure advertising against a label we did not produce. SponsorBlock [14] is a browser extension whose users mark the sponsor segments of a video so that others can skip them. Its contributors had marked 318 sponsor segments in 292 of these episodes before this work began, and those segments cover 4.7% of what is said in those episodes. That 4.7% is the share of its uses a phrase would spend inside an advertisement by chance. The label supports recall but not precision, because an unmarked stretch is not evidence that it is not an advertisement: contributors mark what they want to skip. Coverage is 37% of episodes and none before 2021, so every label-derived figure describes the later part of the corpus.

3.2 Contamination Removal

The detector is cue-anchored rather than repetition-based. It anchors on promotional formulae such as `brought to you by`, `use code` and `percent off`, and on brand tokens learned from the corpus, and treats a window around each anchor as an advertisement. A token enters the brand vocabulary only when it is absent from general English, because a first version that admitted merely rare words removed genuine scientific vocabulary along with the brands. The window is eight minutes of speech, 240 s either side of the cue, because the human-marked segments have a median length of 423 s with the cue anywhere inside them; past 240 s, a grid search shows, each further point of brand removal costs disproportionately more genuine vocabulary. The detector removes 9.6% of the corpus, 8.0 pp of it flagged as sponsor reads and the rest as boilerplate such as intros and sign-offs, event promotion, and clips; 90.4% is kept.

3.3 Term Discovery

Four detectors run over the cleaned corpus and their outputs are unioned. Each over-captures by design, because precision is bought in the hand-labelling step below. D1 fires on metalinguistic coining markers, surface patterns in which a speaker announces a term as they invent it (*what I call*, *a term I coined*), following [9]; it is the only detector that can find a term used once and then dropped. D2a selects rare multi-word phrases, with a phrase's rarity taken as the Zipf frequency of its rarest content word on the Zipf scale [18] as implemented in `wordfreq` [17], on which everyday words sit above 4 and we require 3.0 or below. D2b selects phrases with high collocation strength, PMI of 8.0 or more, because coinages built from ordinary words, such as `blue team`, are not rare and can be found only by cohesion. D3 selects rare single words at Zipf 2.5 or below. For a phrase $w_1 \dots w_n$ we compute PMI as $\log_2 p(\text{phrase}) - \sum_i \log_2 p(w_i)$, without length normalisation; each added word subtracts another term, so PMI rises with phrase length, which Section 4.1 controls for. A term must occur in at least three episodes to enter the candidate set, except when D1 fires, which admits a term on its first and only use. That exemption is what makes stillborn coinages findable at all, and Section 4.3 says what it does to one figure. The union yields 255,690 candidates.

3.4 Hand Labelling

A score combining rarity and collocation strength, with marker-detected terms admitted unconditionally, cut the candidates to a 700-term shortlist, because a quarter of a million candidates cannot be read. The author labelled the shortlist by hand on two questions: where the term came from (`origin`: coined here, borrowed from a technical field, ordinary language, or a transcription artifact) and how it is used (`use`: literally, to frame a claim, or both). Each row showed the term, its first use in context and the number of episodes it appears in, but nothing about whether it later caught on, so the labels are blind to a term's fate.

4 Results

4.1 The Ranking Result

On the cleaned corpus, after the detector of Section 3.2 has run, the five phrases collocation ranks highest are all sponsor copy, three of them variants of one mattress advertisement, led by `midnight luxe hybrid mattress won` (PMI 66.3) and `tj robinson's fresh pressed olive` (66.0). All five sit far above

the podcast's highest-scoring coined term, `time traveling money printer`, at 36.2.

Measured against the SponsorBlock label, the median term in the top 100 by collocation spends every one of its uses inside a marked segment, and 60% of them spend more than half (Figure 1). The median is over the 87 of the 100 that occur in a labelled episode; the other 13 are not counted. Ranked by rarity, 97 of the top 100 are attested and the median is 0%, as it is for the 46 hand-labelled coinages attested there. PMI rises with phrase length (Spearman $\rho = 0.56$) and 99 of the top 100 are five words long, so the controls are random five-word phrases, matched or not to the top 100 on how often they occur. Both sit at a median of 0%, with 3 to 4% above half, which is what a 4.7% base rate predicts for phrases seen once or twice, as most controls are. A top-collocation phrase is 17.8 times as likely to sit mostly inside marked advertising as a pooled control phrase (52 of 87 against 11 of 328, pooling the matched control in Figure 1 with a second, unmatched random set of 170; 95% confidence interval (CI) [9.7, 32.7]).

What this does not show. Whether either signal *separates* advertising from terminology across their whole distributions is a different question, and its answer depends on the comparison class. A signal can be captured by advertising at its top and useless as a classifier overall, and only the first matters to a pipeline that takes the top of a list. We scored the 26 well-attested coinages against every 1–5-word phrase inside the eight-minute window around a promotional cue, which we call the ad-window class ($n = 52{,}930$ phrase types occurring at least five times). That class is ordinary speech rather than advertising vocabulary: 95.9% of it contains none of the 118 brand words the detector learned. AUC, the area under the receiver operating characteristic (ROC) curve, is here the chance that a randomly picked coinage outscores a randomly picked phrase from the class. We report it undirected, as $\max(a, 1-a)$, because we had no prior about the direction; it therefore cannot fall below 0.5, and chance alone produces 0.545 at $n = 26$. Rarity reads 0.647 and collocation 0.545 against that class, neither a separation. Against the 2,177 phrases in the window that do carry a learned brand word, 2,064 of them with a defined PMI, collocation reads 0.605 (95% CI [0.51, 0.71], permutation $p = 0.07$), and rarity reads 0.981, a tautology, because brand membership is defined by absence from general English. No comparison class available to us tests separation cleanly, which is why the claim of this paper is about ranking. The test covers 26 of the 64 coinage rows, because scoring a term needs a minimum number of occurrences and that excludes the stillborn coinages.

4.2 Detector Cost

Cleaning is not free, and both of its error rates are reported as results (Table 1). *Recall* is how much of the real advertising the detector finds: 84.9% of the crowd-marked speech by utterance count (95% CI [84.4, 85.4]; 86.5% by duration), against 17.3% (24.5% by duration) for a baseline that removes only verbatim repetition, a ratio of 4.9. A blind check of 50 kept utterances found one advertisement (Table 1). Recall is an in-sample fit: the eight-minute window was chosen by a grid search over the same 318 segments it is scored against, so the true figure is lower. *Precision* is 47.4%. That figure corrects the author's raw blind judgement, 37.5% (33/88) of removed utterances judged to be advertising, for the fact that the judge is an imperfect detector: measured against the SponsorBlock marks, the judge recognised advertising inside marked segments 70.3% of the time (26 of 37) and called speech outside them advertising 7.9% of the time (5 of 63). We correct for both errors, treating unmarked speech as non-advertising, which Section 3.1 warns overstates the second correction and makes 47.4% a lower bound; the interval is wide, 95% CI [28.0, 71.7], because the calibration rests on 37 marked utterances. At that precision, 8.0 pp times one minus 0.474, roughly 4.2 pp of the corpus is genuine content removed with the advertising (95% CI [2.3, 5.8]). The two cautions cut in opposite directions: in-sample recall overstates the catch, so more advertising survives cleaning than the figure implies, while over-removal thins the pool the advertising is ranked against, which bounds the size of the effect but not its sign.

Table 1: Contamination detector cost. Recall is an in-sample fit (Section 4.2); by duration, recall reads 86.5% and the baseline 24.5%. Intervals are percentage points: Wilson for proportions, parametric bootstrap for the calibrated row. Bold marks the precision figure the paper uses.

	value	95% CI
recall vs. 318 segments, in-sample	84.9%	[84.4, 85.4]
repetition-only baseline	17.3%	[16.8, 17.9]
boilerplate precision	100% (8/8)	[67.6, 100.0]
sponsor precision, raw	37.5% (33/88)	[28.1, 47.9]
calibrated, two-sided	**47.4%**	[28.0, 71.7]
false-negative rate, kept utterances	2.0% (1/50)	[0.4, 10.5]

4.3 The Lexicon

The lexicon contains 59 distinct coinages across 64 surface rows, with transcription and inflectional variants aggregated. Twenty-four of the 59 (40.7%) are stillborn, used in the episode that introduced them and never again. That figure is a floor set by the discovery route rather than an estimate of how often a coinage dies: a term used once and not announced on air cannot be discovered at all, and every one of the 24 was found by the marker detector, the only route exempt from the three-episode threshold. The full lexicon is released with the paper.

5 Discussion

Why the two signals disagree. A sponsor read is written once and then performed dozens of times, word for word. That is exactly what collocation rewards, so advertising reaches the top of the collocation ranking ahead of anything a speaker coined in conversation. Rarity behaves the other way round, and the cleaning step is why. Brand names look extremely rare because they do not occur in ordinary English, and that absence is the very rule the detector uses to learn them, so cleaning strips out the words that would otherwise top the rarity ranking. What cleaning does not strip out is the sponsor's phrasing where it drifts outside a detected block, and phrasing is what collocation measures.

The problem appeared in this project's own pipeline. An early shortlist for this paper, ranked with these same two signals, behaved exactly as described here: leftover advertising copy outranked genuine coinages, and two of the podcast's best-known terms, `omega principle` and the short form `money printer`, dropped off the list. What brought them back into discovery was not a better advertising filter but fixing three defects in the marker detector, after which the known-answer check reads 12

of 12 at discovery. Both terms are lost again at the shortlist cut (Section 6), a separate failure we have not fixed.

How reliable the labels are. All hand labelling was done by one person, so a language model re-labelled all 700 rows blind and the two sets were compared. On the four-way `origin` split they agree at Cohen's $\kappa = 0.271$, which is poor; on the binary question the lexicon rests on, coined or not, at $\kappa = 0.447$, moderate at best, with the model recovering 75% of the hand-marked coinages.

6 Limitations

The main corpus is one podcast, two speakers and one subject area, and its advertising label covers 37% of its episodes and none before 2021. Of the paper's two products, the ranking asymmetry can be tested elsewhere and the lexicon cannot. The asymmetry needs a sponsored podcast with SponsorBlock coverage, the same cleaner and the two rankings, so we replicated it on eight other podcasts: a physics channel, a science outreach show, a health-science show, a wellness show, a business show and three political interview shows, between 50 and 316 episodes and 35 to 129 hours each. On seven of the eight, the median phrase in the top 100 by collocation spends 100% of its uses inside marked advertising. On the eighth, the health-science show, the eight of the top ten that occur in a labelled episode are all sponsor reads. But the top 100 also holds the host's own opening and closing lines, performed word for word in every episode, which the label marks as the host's own promotion rather than as advertising, and some genuine neuroscience terms, so the median there is 5%, or 68% with the host's promotions counted. On three shows, the physics channel and two of the political shows, nearly every five-word phrase that repeats across episodes is advertising, so the control phrases score as high as the top 100 and those three results say more about the pool than the ranking. The business, science outreach, wellness and remaining political show test the ranking cleanly, and all four replicate. On the rarity side, the top of the ranking on every show is a brand name or web address that survived cleaning. Where many survived, on the physics channel and the three political shows, the median of the top 100 reads 100% (four shows); where few did, on the business, science outreach and wellness shows, it reads 0% (three shows); on the health-science show it reads 8%. So the full asymmetry, collocation at 100% and rarity at 0% with the control at 0%, replicates on three shows. Rarity finds advertising wherever cleaning has left the brand names in, which is the mechanism of Section 5 from the other side. The lexicon does not travel, because judging whether a phrase is a coinage of these hosts requires knowing their vocabulary, and no second human annotator checked ours.

Against twelve terms fixed before the run, all of which survive discovery, the shortlist keeps 8 of 12; among the four it drops is `omega principle`, the hosts' best-known coinage, so it is missing from the released lexicon.

7 Conclusion

Advertising on a sponsored corpus is not noise for term extraction but a competing class, and the two standard signals treat it in opposite ways once a cleaner has done its work: collocation ranks what survives first, and rarity does not once the cleaner has taken the brand names out. From that point, which signal a pipeline ranks by decides how much advertising its term list inherits; before it, both deliver it. The mechanism is independent of the corpus by construction, because a script performed verbatim is a fixed expression in any corpus; whether the effect is as large elsewhere is what the replication in Section 6 tests. A practitioner who ranks by collocation on sponsored speech should check the top of the list against an advertising label before reading it as terminology.

References

[1] Khurshid Ahmad, Lee Gillam, and Lena Tostevin. 1999. University of Surrey participation in trec8: weirdness indexing for logical document extrapolation and retrieval (wilder). In *The Eighth Text REtrieval Conference (TREC 8)*. NIST Special Publication 500-246, 717.

[2] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43, 3, 209–226. DOI: 10.1007/s10579-009-9081-4.

[3] Marco Baroni, Francis Chantree, Adam Kilgarriff, and Serge Sharoff. 2008. Cleaneval: a competition for cleaning web pages. In *Proceedings of the Language Resources and Evaluation Conference*. DOI: 10.63317/4y7ib5ikczny.

[4] Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16, 1, 22–29. https://aclanthology.org/J90-1003/.

[5] Ann Clifton et al. 2020. 100,000 podcasts: a spoken English document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, 5903–5917. DOI: 10.18653/v1/2020.coling-main.519.

[6] Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19, 1, 61–74.

[7] Ingrid Falk, Delphine Bernhard, and Christophe Gérard. 2014. From non word to new word: automatically identifying neologisms in French newspapers. In *Proceedings of the Language Resources and Evaluation Conference*, 4337–4344. DOI: 10.63317/3tnyws3xnn2a.

[8] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the c-value/nc-value method. *International Journal on Digital Libraries*, 3, 2, 115–130. DOI: 10.1007/s007999900023.

[9] Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics -*. Vol. 2, 539. DOI: 10.3115/992133.992154.

[10] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J. F. Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. 2021. Trec 2020 podcasts track overview. arXiv:2103.15953. (2021). DOI: 10.48550/arxiv.2103.15953.

[11] Rosie Jones et al. 2021. Current challenges and future directions in podcast information access. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1554–1565. DOI: 10.1145/3404835.3462805.

[12] Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6, 1, 97–133. DOI: 10.1075/ijcl.6.1.05kil.

[13] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, 441–450. DOI: 10.1145/1718487.1718542.

[14] Ajay Ramachandran and the SponsorBlock contributors. 2026. Sponsorblock: skip sponsorships on youtube. Browser extension and crowdsourced segment database. https://sponsor.ajay.app, source at https://github.com/ajayyy/SponsorBlock. Segments retrieved 28 July 2026. (2026).

[15] Sravana Reddy, Yongze Yu, Aasish Pappu, Aswin Sivaraman, Rezvaneh Rezapour, and Rosie Jones. 2021. Detecting extraneous content in podcasts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1166–1173. DOI: 10.18653/v1/2021.eacl-main.99.

[16] Diego Rossini and Lonneke van der Plas. 2026. From 124 million tokens to 1,021 neologisms: a large-scale pipeline for automatic neologism detection. arXiv:2605.06426. Accepted at the NeoLLM 2026 workshop, co-located with LREC-COLING 2026. (2026). DOI: 10.48550/arxiv.2605.06426.

[17] Robyn Speer. 2022. Wordfreq: a python library for looking up the frequencies of words in many languages. Version 3.0.2, Zenodo. Version 3.1.1 used here. (2022). DOI: 10.5281/zenodo.7199437.

[18] Walter J. B. van Heuven, Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. SUBTLEX-UK: a new and improved word frequency database for British English. *Quarterly Journal of Experimental Psychology*, 67, 6, 1176–1190. DOI: 10.1080/17470218.2013.850521.

Indeks avtorjev / Author index

Achuthan Krishnasree 146
Bobnar Lovro 117
Brank Janez 11
Brence Jure 69
Brodschneider Aleksandar 113
Brunet Pau 77
Calcina Erik 27, 31, 35, 73
Cholakova Simona 89
Čibej Jaka 101
Đuranović Vuk 55
Fučka Matic 93
Gams Matjaž 113
Grobelnik Marko 11, 19, 61, 150
Gryllias Konstantinos 85
Guruvayur Sivaramakrishnan R 146
Hosseini Seyed Iman 85
Hrovatin Niki 122
Jeršek Domen 7
Kažič Blaž 7
Kenda Klemen 7, 39, 105, 109
Klančič Rok 39
Klun Žiga 69
Kocev Dragi 69
Kofilovski Stefan 142
Kokalj Nina 35
Komarova Nadezhda 51
Kralj Novak Petra 126
Kreft Gašper 130
Leskovec Gašper 109
Liang Yunfei 93
Macak Martin 150
Mammana Dario 126
Mečkaroska Natalija 23
Mladenić Dunja 11, 15, 19, 27, 61, 81, 85
Mladenić Grobelnik Adrian 47
Motamedi Elham 65
Mylonas Costas 109
Natev Andrej 65
Novak Erik 31, 35, 73
Novalija Inna 11, 51, 65
Osojnik Aljaž 130, 134
Pesjak Drejc 15, 81
Petek Luka 154
Piiarska Kristina 77
Pita Costa Joao 138, 146
PMacak MartinI1 158
Podgorelec Vili 154
Praprotnik Kaja 122
Rei Luis 81
Robnik-Šikonja Marko 55
Rupnik Jan 15
Seghesio Lorenzo 85
Shpanko Nadiia 73
Sittar Abdul 31, 35
Skočaj Danijel 93

Škrlj Blaž....19
Srakar Andrej....97, 138
Šturm Jan....138
Telesca Luciano....27
Topal Oleksandra....27, 138
Tošić Aleksandar....77, 89
Trajkov Georgi....19, 23, 43, 81, 142
Uku Sidorela....105
Vake Domen....89, 122
Vičič Jernej....122
Vokač Jakob....61
Ženko Bernard....130, 134
Zmazek Gal....134
Žnidaršič Martin....130, 134