

Zbornik 22. mednarodne multikonference  
**INFORMACIJSKA DRUŽBA - IS 2019**  
Zvezek C

Proceedings of the 22nd International Multiconference  
**INFORMATION SOCIETY - IS 2019**  
Volume C

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

Uredila / Edited by

Dr.inja Mladenić Marko Grubelnik

<http://is.ijs.si>


7 October 2019 / 7 October 2019  
Ljubljana, Slovenia

Urednika:

Dunja Mladenić  
Artificial Intelligence Laboratory  
Jožef Stefan Institute, Ljubljana

Marko Grobelnik  
Artificial Intelligence Laboratory  
Jožef Stefan Institute, Ljubljana

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

Na naslovnici je uporabljena slika robota podjetja 

Dostop do e-publikacije:  
<http://library.ijs.si/Stacks/Proceedings/InformationSociety>

Ljubljana, oktober 2019



# PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2019

Multikonferenca Informaci družba (<http://is.ijs.si>) je z dvaindvajseto zaporedno prireditvijo tradicionalni osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Informacijska družba, znanje in umetna inteligenca so - in to čedalje bolj – nosilci razvoja človeške civilizacije. Se bo neverjetna rast nadaljevala in nas ponesla v novo civilizacijsko obdobje? Bosta IKT in zlasti umetna inteligenca omogočila nadaljnji razcvet civilizacije ali pa bodo demografske, družbene, medčloveške in okoljske težave povzročile zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so notranji in zunanji konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali 12 odličnih neodvisnih konferenc. Zajema okoli 200 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditve bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica (<http://www.informatica.si/>), ki se ponaša z 42-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2019 sestavljajo naslednje samostojne konference:

- 6. študentska računalniška konferenca
- Etika in stroka
- Interakcija človek računalnik v informacijski družbi
- Izkopavanje znanja in podatkovna skladišča
- Kognitivna znanost
- Kognitonika
- Ljudje in okolje
- Mednarodna konferenca o prenosu tehnologij
- Robotika
- Slovenska konferenca o umetni inteligenci
- Srednje-evropska konferenca o uporabnih in teoretičnih računalniških znanostih
- Vzgoja in izobraževanje v informacijski družbi

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, SLAIS, DKZ in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in institucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V 2019 bomo sedmič podelili nagrado za življenjske dosežke v čast Donalda Michieja in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel [REDACTED]. Priznanje za dosežek leta bo pripadlo [REDACTED]. Podeljujemo tudi nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je [REDACTED], jagodo pa [REDACTED]. Čestitke nagrajencem!

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

# FOREWORD - INFORMATION SOCIETY 2019

The Information Society Multiconference (<http://is.ijs.si>) is the traditional Central European event in the field of information society, computer science and informatics for the twenty-second consecutive year. Information society, knowledge and artificial intelligence are - and increasingly so - the central pillars of human civilization. Will the incredible growth continue and take us into a new civilization period? Will ICT, and in particular artificial intelligence, allow civilization to flourish or will demographic, social, and environmental problems stifle growth? More and more indicators point to both extremes - that we are moving into the next civilization period, and at the same time the internal and external conflicts of modern society are becoming increasingly difficult to manage.

The Multiconference is running parallel sessions with 200 presentations of scientific papers at twelve conferences, many round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal with its 42-years tradition of excellent research publishing.

The Information Society 2019 Multiconference consists of the following conferences:

- 6. Student Computer Science Research Conference
- Professional Ethics
- Human – Computer Interaction in Information Society
- Data Mining and Data Warehouses
- Cognitive Science
- International Conference on Cognitonics
- People and Environment
- International Conference of Transfer of Technologies – ITTC
- Robotics
- Slovenian Conference on Artificial Intelligence
- Middle-European Conference on Applied Theoretical Computer Science
- Education in Information Society

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, SLAIS, DKZ and the second national engineering academy, the Slovenian Engineering Academy. In the name of the conference organizers, we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the fifteenth year, the award for life-long outstanding contributions will be presented in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to [REDACTED] for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to [REDACTED]. The information lemon goes to [REDACTED]. The information strawberry is awarded [REDACTED]. Congratulations!

Mojca Ciglarič, Programme Committee Chair  
Matjaž Gams, Organizing Committee Chair



# KONFERENČNI ODBORI

## CONFERENCE COMMITTEES

### *International Programme Committee*

Vladimir Bajic, Južna Afrika  
Heiner Benking, Nemčija  
Se Woo Cheon, Južna Koreja  
Howie Firth, Škotska  
Olga Fomichova, Rusija  
Vladimir Fomichov, Rusija  
Vesna Hljuz Dobric, Hrvaška  
Alfred Inselberg, Izrael  
Jay Liebowitz, ZDA  
Huan Liu, Singapur  
Henz Martin, Nemčija  
Marcin Paprzycki, ZDA  
Claude Sammut, Avstralija  
Jiri Wiedermann, Češka  
Xindong Wu, ZDA  
Yiming Ye, ZDA  
Ning Zhong, ZDA  
Wray Buntine, Avstralija  
Bezalel Gavish, ZDA  
Gal A. Kaminka, Izrael  
Mike Bain, Avstralija  
Michela Milano, Italija  
Derong Liu, Chicago, ZDA  
Toby Walsh, Avstralija

### *Organizing Committee*

Matjaž Gams, chair  
Mitja Luštrek  
Lana Zemljak  
Vesna Koricki  
Marjetka Šprah  
Mitja Lasič  
Blaž Mahnič  
Jani Bizjak  
Tine Kolenik

### *Programme Committee*

Mojca Cigliarič, chair  
Bojan Orel, co-chair  
Franc Solina  
Viljan Mahnič  
Cene Bavec  
Tomaž Kalin  
Jozsef Györköös  
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  
Vladislav Rajkovič  
Grega Repovš  
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  
Ivan Rozman  
Niko Schlamberger  
Stanko Strmčnik  
Jurij Šilc  
Jurij Tasič  
Denis Trček  
Andrej Ule  
Tanja Urbančič  
Boštjan Vilfan  
Baldomir Zajc  
Blaž Zupan  
Boris Žemva  
Leon Žlajpah



## KAZALO / TABLE OF CONTENTS

<b><i>Izkopavanje znanja in podatkovna skladišča (SiKDD) / Data Mining and Data Warehouses (SiKDD)</i></b> .....	<b>1</b>
PREDGOVOR / FOREWORD .....	3
PROGRAMSKI ODBORI / PROGRAMME COMMITTEES .....	4
Preferences of Users on Cross-Site OER Recommendations: Stay or Leave? / Sunar Ayşe Saliha, Novak Erik, Urbančič Jasna, Mladenić Dunja .....	5
The Next Big Thing In Science / Mladenić Grobelnik Adrian, Mladenić Dunja, Grobelnik Marko .....	9
Deep Language Classification for Relabeling of Financial News and its application in Stock Price Forecasting / Trichilo Giulio, Torkar Miha .....	13
Semantic Enrichment and Analysis of Legal Domain Documents / Massri Beshir M., Brezec Sara, Novak Erik, Kenda Klemen .....	17
Health News Bias and Epidemic Intelligence for Public Health / Pita Costa Jao, Fuat F., Stopar L., Grobelnik Marko, Mladenić Dunja, Košmerlj Aljaž, Belayeva E., Rei L., Leban G. ....	21
Latent distance graphs from news data / Bizjak Luka, Torkar Miha, Košmerlj Aljaž .....	25
Document Embedding Models on Environmental Legal Documents / Kralj Samo, Urbančič Živa, Novak Erik, Kenda Klemen .....	29
Local-to-global analysis of influenza-like-illness data / Pita Costa Jao, Fuat F., Stopar L., Paolotti D., Hirsch M., Mexia R. ....	33
Feature Selection in Land-Cover Classification using EO-learn / Koprivec Filip, Peternelj Jože, Kenda Klemen .....	37
Identifying events in mobility data / Kavšek Branko, Mladenić Dunja, Malik Omar, Szymanski Boleslaw K. ....	41
Early land cover classification with Sentinel 2 satellite images and temperature data / Čerin Matej, Koprivec Filip, Kenda Klemen .....	45
How overall coverage of class association rules affects the accuracy of the classifier? / Mattiev Jamolbek, Kavšek Branko .....	49
Epileptic Seizure Detection Using Topographic Maps and Deep Machine Learning / Kojanec Patrik, Kavšek Branko, Teixeira César A. D. ....	53
Demand Forecasting for Industry 4.0: predicting discrete demand from multiple sources for B2B domain / Rožanec Jože Martin, Mladenić Dunja, Fortuna Blaž .....	57
Empirical study on the performance of Neuro Evolution of Augmenting Topologies (NEAT) / Vake Domen, Tošić Aleksandar, Vičič Jernej .....	61
Learning Hand-Eye Coordination on NAO and its Applications / Boc Ana Gaja, Bertonecelj Čadež Sara .....	65
<b><i>Indeks avtorjev / Author index</i></b> .....	<b>69</b>



Zbornik 22. mednarodne multikonference  
**INFORMACIJSKA DRUŽBA - IS 2019**  
Zvezek C

Proceedings of the 22nd International Multiconference  
**INFORMATION SOCIETY - IS 2019**  
Volume C

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

Uredila / Edited by

Dr.inja Mladenitć Marko Grubelnik

<http://is.ijs.si>

7 October 2019 / 7 October 2019  
Ljubljana, Slovenia



## PREDGOVOR

Tehnologije, ki se ukvarjajo s podatki so v devetdesetih letih 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, prišlo je do standardizacije procesov, povpraševalnih jezikov itd. 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 – pojavilo se je t.i. skladiščenje podatkov (data warehousing), ki je postalo standarden del informacijskih sistemov v podjetjih. Paradigma OLAP (On-Line-Analytical-Processing) zahteva od uporabnika, da še vedno sam postavlja sistemu vprašanja in dobiva nanje odgovore in na vizualen način preverja in išče izstopajoče situacije. Ker seveda to ni vedno mogoče, se je pojavila potreba po avtomatski analizi podatkov oz. z drugimi besedami to, da sistem sam pove, kaj bi utegnilo biti zanimivo za uporabnika – to prinašajo tehnike odkrivanja znanja v podatkih (data mining), ki iz obstoječih podatkov skušajo pridobiti novo znanje in tako uporabniku nudijo novo razumevanje dogajanj zajetih v podatkih. Slovenska KDD konferenca pokriva vsebine, ki se ukvarjajo z analizo podatkov in odkrivanjem znanja v podatkih: pristope, orodja, probleme in rešitve.

## FOREWORD

Data driven technologies have significantly progressed after mid 90's. 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. At this point, data warehousing with On-Line-Analytical-Processing entered as a usual part of a company's information system portfolio, requiring from the user to set well defined questions about the aggregated views to the data. Data Mining is a technology developed after year 2000, offering automatic data analysis trying to obtain new discoveries from the existing data and enabling a user new insights in the data. In this respect, the Slovenian KDD conference (SiKDD) covers a broad area including Statistical Data Analysis, Data, Text and Multimedia Mining, Semantic Technologies, Link Detection and Link Analysis, Social Network Analysis, Data Warehouses.

## **PROGRAMSKI ODBOR / PROGRAMME COMMITTEE**

Janez Brank, Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana

Marko Grobelnik, Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana

Branko Kavšek, University of Primorska, Koper

Aljaž Košmerlj, Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana

Dunja Mladenić, Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana

Inna Novalija, Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana



# Preferences of Users on Cross-Site OER Recommendations: Stay or Leave?

**Ayşe Saliha Sunar**  
assunar@beu.edu.tr  
Jozef Stefan Institute  
Ljubljana, Slovenia  
Bitlis Eren University  
Bitlis, Turkey

**Jasna Urbančič**  
jasna.urbancic@ijs.si  
Jozef Stefan Institute  
Ljubljana, Slovenia

**Erik Novak**  
erik.novak@ijs.si  
Jozef Stefan Institute  
Jozef Stefan International  
Postgraduate School  
Ljubljana, Slovenia

**Dunja Mladenici**  
dunja.mladenici@ijs.si  
Jozef Stefan Institute  
Jozef Stefan International  
Postgraduate School  
Ljubljana, Slovenia

## ABSTRACT

In education we can find different open educational resource (OER) providers that are serving resources in different modalities, formats and languages. These providers can be the actual resource creators or re-distributors that redirect the user to the actual provider. In recent work, we developed a recommendation engine which provides content-based recommendations from multiple resource providers, enabling the users to navigate between the providers and their resources. In this paper, we investigate the users' choice on the recommended items focusing on the cross-site user learning activities. The results show that the users tend to stay on the same website and not choose the first item in the recommendation list.

## CCS CONCEPTS

• **Information systems** → *Content ranking*.

## KEYWORDS

open educational resources, recommendation system, cross-site recommendations, learning analytics, data visualization

## ACM Reference Format:

Ayşe Saliha Sunar, Erik Novak, Jasna Urbančič, and Dunja Mladenici. 2019. Preferences of Users on Cross-Site OER Recommendations: Stay or Leave?. In *SiKDD 2019*. ACM, New York, NY, USA, 4 pages.

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).

Slovenian KDD Conference, October 10, 2019, Ljubljana, Slovenia

© 2019 Copyright held by the owner/author(s).

## 1 INTRODUCTION

Open Educational Resources (OERs), as defined by the UNESCO,<sup>1</sup> are teaching, learning and research materials, digital or otherwise, that reside in the public domain or have been released under an open license that permits no-cost access, use, adaptation and redistribution by others with no or limited restrictions. As such, digital OERs have many advantages over traditional learning materials. Namely, OERs reduce the costs for the students, the dissemination of information is faster, and the resources can be accessed from everywhere. However, there are also some reservations, such as quality and reliability of the materials, and intellectual property rights ownership. Additionally, an OER user faces a very fragmented landscape of repositories containing OERs, which makes finding relevant OERs a difficult task for both students and teachers.

Therefore, we aim to connect the scattered OERs by enriching the material with additional semantic information, automatic and machine translation, as well as providing services for cross-site recommendations to make finding appropriate OERs easier. Currently, the repositories are highly specialised in terms of scientific domains, content type, level of education, and language. From the perspective of a student, the student may have to search OERs in different repositories for each class which is an undesirable situation. Such search is inefficient and time consuming, it also leads to sub-optimal search results and a negative user experience with OERs.

In this paper, we aim to provide some insight into the preferences of the users regarding cross-site recommendations. To discover users' preferences, we analyse user transition

<sup>1</sup>Definition adopted from <https://en.unesco.org/themes/building-knowledge-societies/oer>.

data from content-based cross-site recommender engine embedded into two OERs repositories. We focus on the cross-events, which happen when the user selects OER materials from a different repository from the list of recommendations. The findings are the first step in evaluating how users perceive such recommendations, how good the recommender is, and to further improve the recommendations.

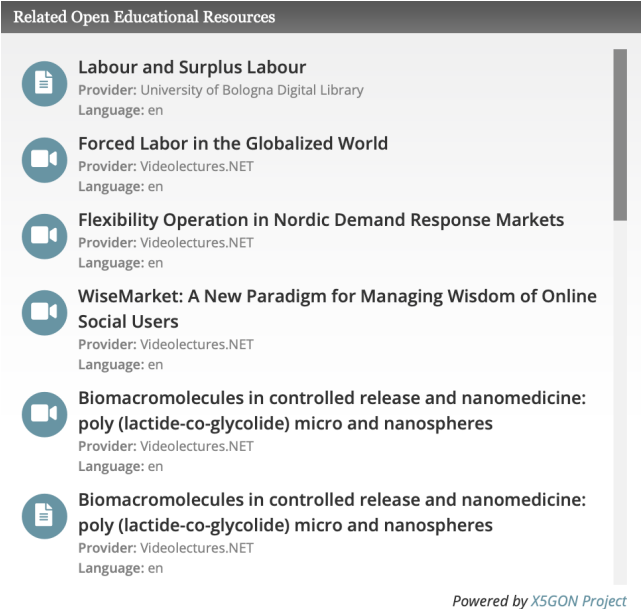
## 2 RELATED WORK

Recommender engines produce the results which are identified as the most relevant to the user by using different methods such as content-based or collaborative filtering. Considering the personal differences such as interest, study goal, time restriction, and capabilities of owned devices, not all users would choose the first ranked item in recommendations. For example, Hajri et al [4] created a recommendation engine to support MOOC learners with complementary OERs. Their study identifies that some learners did not prefer to use external resources to not disperse themselves. This result implies that user may not like to navigate cross-site even though they are provided with an item recommendation very related to their interest. Thus, understanding users' preferences on recommended items is very crucial for maintenance and improvement of recommender engines [2] so that personalised items could be provided to meet each user's preferences [1]. This paper is designed to understand users' cross-site navigation through recommendations on OER and their choice of the item to study.

## 3 CROSS-SITE RECOMMENDATIONS OF OPEN EDUCATIONAL RESOURCES

In order to provide cross-site and cross-language recommendations, we have developed a content-based recommender system which recommends resources based on the similarity of their content. We propose semantic representation of the resources based on Wikipedia concepts, i.e. Wikipedia pages which were identified and linked to particular parts of the material's content. Wikipedia concepts are then used to represent the resources of different modalities - providing a "concept" overview of the resource's content. This semantic representation allows comparing and calculating the similarity between resources of different languages, thus enabling recommendations containing resources in different languages. The recommender system design is described in [5].

Our recommendations are designed to provide the most similar resources based on the user query. The query can be either a) a link to another resource or b) a free-form text. For this paper, we focus on the first option, where the user provides a link to another resource. The recommendations are ordered by resource's similarity to the provided query, i.e. more similar resources appear higher on the list.



**Figure 1: Snapshot of the recommendation window on videolectures.net resource (date: 25.07.2019).**

The recommender system has been provided as a service with a public application programming interface (API).<sup>2</sup> In addition, we have also provided a plugin which allows a website to include the list of the OER recommendations given a query. Figure 1 shows an example of the recommender plugin output on the videolectures.net repository<sup>3</sup>.

The data about users' transitions between the OER materials and recommendations is stored as csv files (we retrieved it on: 04.07.2019) Please note that we only have the transitions directed from www.VideoLectures.net (VL) and www.upv.es (Universitat Politècnica de València - UPV) due to the data sharing policies. We have implemented learning analytics techniques to analyse and visualise the users' preferences on recommendations by using the Python programming language.

## 4 ANALYSIS ON USERS' PREFERENCES

In the transitions dataset we have 233,221 transitions showing the users navigating from one page to another through the recommended items. Please note that we only have the transitions directed from VL and UPV due to the data sharing policies. The data used in our experiments was retrieved on July 4th, 2019.

The users are usually provided with around 20 items in the recommendation list. The data shows that the users tend to choose the item ranked 8.89. Usually, only 5 to 7 items

<sup>2</sup>Documentation is available at <https://platform.x5gon.org/documentation>

<sup>3</sup><http://videolectures.net/>

**Table 1: Frequency of Navigation amongst OER repositories**

Directed from	Directed to	Frequency
VL	VL	176,594 (76%)
VL	UPV	14,212 (6%)
VL	UOS	553 (0.2%)
VL	Nantes	8 (0.0034%)
VL	MIT	8,854 (3.8%)
VL	Bologna	32,882 (14%)
UPV	VL	14 (0.006%)
UPV	UPV	92 (0.04%)
UPV	UOS	0 (0%)
UPV	Nantes	0 (0%)
UPV	MIT	0 (0%)
UPV	Bologna	12 (0.005%)

could fit into the recommendation window, which means that the users tend to scroll down in the recommendation window rather than click on the first recommended item.

According to the statistics, the users have chosen an item from the first page 88914 times (38%) and have scrolled down to chose an item 144,291 times (62%) in the case that 6 items shown at once in the recommendation window (see Fig.1).

### Navigation amongst OER sites

Since the recommendations are cross-site, it is possible for the users to move from one OER repository to another. Because of the data sharing policy among the providers, we can track the transitions from VL and UPV to any partner providers. The sankey diagram in Figure 2 shows the navigation amongst the OER repositories. Table 1 shows the exact number how many times a user is directed from one repository to another.

Apart from the users' decision to choose an item from different domains, the number of items recommended by domain could have an effect on the users' choices. Figure 3 shows the percentage of recommended items by domains on VideoLectures.net and UPV, respectively. We can see that the providers mostly recommend an item from their own domain.

The results can be summarised as follows:

- When a user is on a material, most probably they choose the next material from the same domain, indicating that they prefer to stay on the same website.
- The users have mostly chosen the next item from the VL, Bologna, UPV and MIT respectively.
- All transitions to Nantes, UOS and MIT were directed from VL.

- There is at least one transition from the VL to each of the OER repository listed while there are no transitions between UPV and UOS, Nantes, and MIT. The reason could be that the content of the resources are not similar on these particular repositories so they are rarely shown to the users.

### Chosen vs. Not Chosen

Even though there surely are reasons for a user not to choose the first ranked item in the recommendation list which cannot be traced in the data, we hypothesise that the users would choose one of the first shown items since they are more similar to the viewed item based on their contents. Therefore, observing the trends with the chosen item by investigating its similarity to the other given recommendations would give us insight into the users' behaviours.

A heat map is a convenient tool to visually show the trends. Figure 4 compares the features of the first item in the recommendation lists and the chosen item by the user.

The features are chosen for comparison are explained below.

**Author.** Author of the material

**Language.** Spoken language in the video or the provided language in the text

**Provider.** The website domain of the material provider e.g. videolectures.net

**Type.** Type of the material e.g. pdf or mp4

**Wikipedia Concepts.** The Wikipedia concepts linked to the resource's content. These were acquired through the Wikifier [3], a web service that finds and links text elements to Wikipedia concepts.

The items ranked first in the list and selected by the users are not included here. In the graph, the more purplish the more different, the more greenish the more similar.

According to the graph, the author information is the least similar feature between the selected item and the first item shown in the recommendation list. This implies that the users rarely chose the item authored by the same author of the first item in the list. It should be noted here that one of the reasons for this could be the lack of materials authored by the same author. It is also observed that language, provider and type are not necessarily the same.

The first ranked item is contentwise the most similar item to the currently viewed item (see Section 3). However, the items chosen by users are not always the most similar items. This result implies that the currently implemented content-based filtering method is not enough to meet the users' preferences, and other aspects must be considered.

## 5 CONCLUSION

The presented research is designed to investigate users' navigation between the different open educational resource

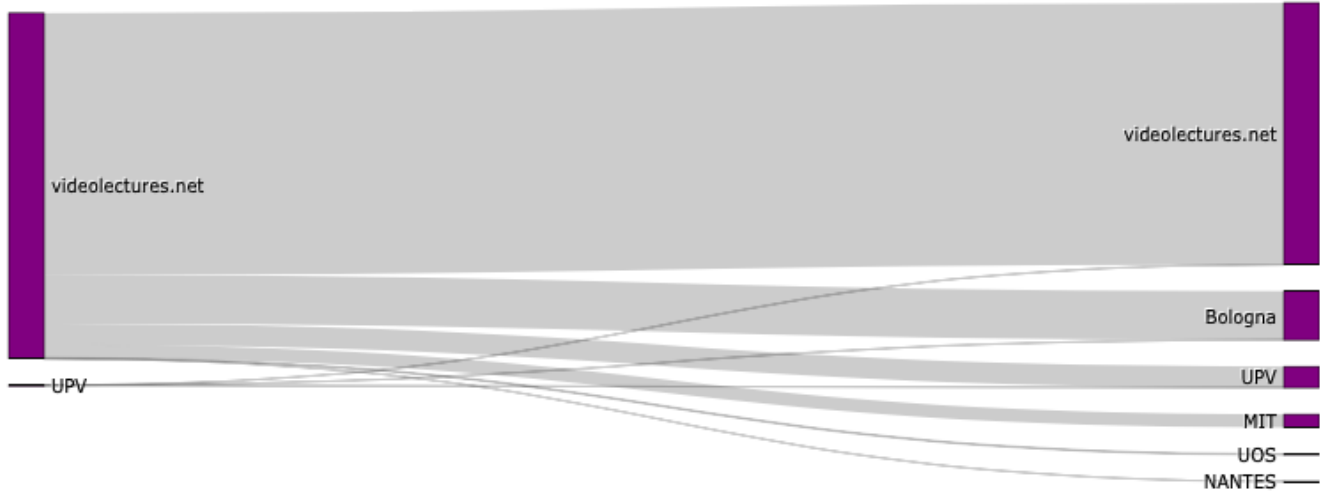


Figure 2: Navigation amongst the OER providers

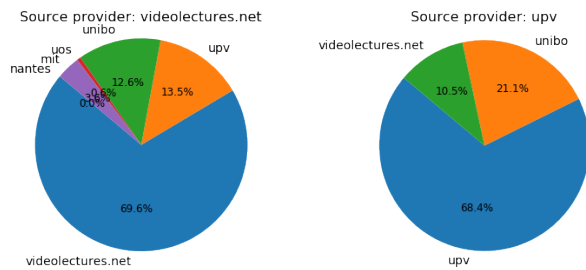


Figure 3: Number of materials recommended by domain

providers through an embedded recommendation engine on their platforms. We observe that the users mostly chose to stay within the same domain provider. Interestingly, we observed that the users did not choose the first couple of items that are ranked higher in the recommendation list, but they rather chose items ranked at around 8th place. This result shed light on users' preferences on cross-site OERs but also pave a way to further research to i) deeper behavioural analysis on user preferences and ii) improve the recommender engine which not only implement a content-based filtering but a method which is modified with personalised attributes.

## 6 ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and X5GON European Unions Horizon 2020 project under grant agreement No 761758.

## REFERENCES

- [1] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *In 29th*

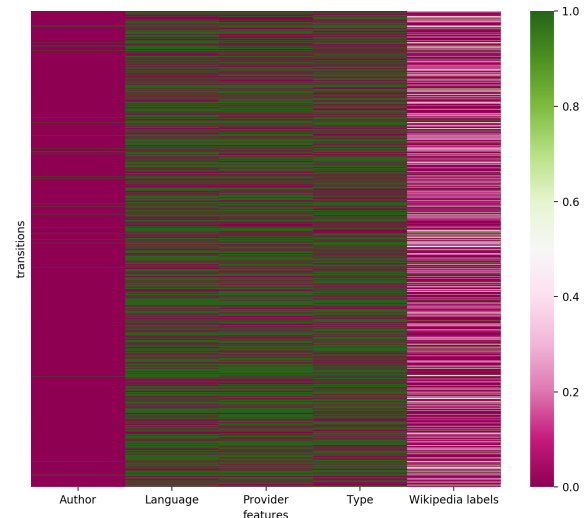


Figure 4: Comparison of the selected item and the first ranked item in the recommendation list

- Int'l ACM SIGIR Conf on Research and Development in Information Retrieval*. ACM, 19–26.
- [2] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. [n.d.]. Learning user interaction models for predicting web search result preferences.
- [3] Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant Wikipedia concepts. In *SiKDD*.
- [4] Hiba Hajri, Yolaine Bourda, and Fabrice Popineau. 2018. Personalized Recommendation of Open Educational Resources in MOOCs. In *International Conference on Computer Supported Education*. Springer, 166–190.
- [5] Erik Novak, Jasna Urbančič, and Miha Jenko. 2018. Preparing multi-modal data for natural language processing. In *SiKDD*.

# The Next Big Thing In Science

Adrian Mladenec Grobelnik  
Artificial Intelligence Laboratory  
Jozef Stefan Institute  
Ljubljana Slovenia  
adrian.m.grobelnik@ijs.si

Dunja Mladenec  
Artificial Intelligence Laboratory  
Jozef Stefan Institute  
Ljubljana Slovenia  
dunja.mladenec@ijs.si

Marko Grobelnik  
Artificial Intelligence Laboratory  
Jozef Stefan Institute  
Ljubljana Slovenia  
marko.grobelnik@ijs.si

## ABSTRACT

This paper presents an approach to predicting the future development of scientific research based on scientific publications from the past two centuries. We have applied machine learning methods on the Microsoft Academic Graph dataset of scientific publications. Our experimental results show that the best performance is obtained for a noticeable increase of the topic frequency in the last 5 years compared to the previous 10 years. In this case, our model achieves precision of 74.3, recall of 71.7 and F1 of 73.0. Some topics that our model identified as promising are: *proton proton collisions, higgs boson, quark, hadron, mobile augmented reality, variable quantum, molecular dynamics simulations, hadronic final states, search for dark matter.*

## CCS CONCEPTS

•[CCS](#) [Information systems](#) [Information retrieval](#) [Document representation](#) [Content analysis and feature selection](#)

## KEYWORDS

Science analysis, machine learning, data representation

## 1 Introduction

With the ever-increasing pace of scientific developments, it is becoming difficult to keep track of current scientific research topics, let alone predict the promising lines for future research. As the quality and quantity of digitized scientific publications is growing, it has enabled modelling the development of scientific publications over time with greater accuracy and efficiency. In our research we explore how a simple Perceptron algorithm performs, given a considerable amount of data.

Our research hypothesis is that scientific topics that will be important in the future, already exist in today's scientific articles. To identify them, we applied machine learning methods on a large database of publications, namely the Microsoft Academic Graph [1]. We have defined a machine learning problem, such that the model predicts early indicators suggesting which scientific topics in today's literature will likely become important in the future.

In related work, researchers have addressed a similar problem also on a part of the Microsoft academic database of publications. They used a binary classifier to predict future developments in science. However, their research was on "Finding rising stars in academia early in their careers" [6]. Their representation comprises of authors' personal and social features. The research presented in [7]

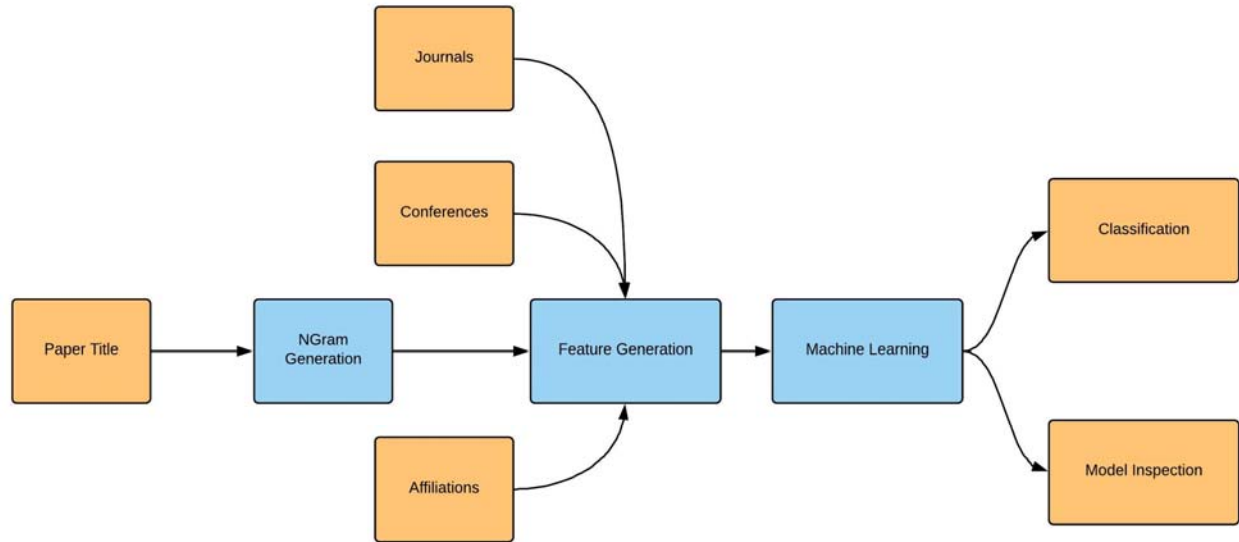
focuses on predicting emerging topics based on citation and co-citation data using clustering methods. The topics are classified to understand the motive forces behind their emergence ("scientific discovery, technological innovation, or exogenous events"). Emerging topics were also addressed in [8] where keywords from MeSH terms of PubMed database are filtered based on their increment rate of appearance in life science publications. In our research, we automatically generate frequent NGrams from the paper titles and use them to construct a machine learning model for predicting which topics will become popular in the future.

The main contributions of this paper are the proposed problem definition, data representation and the identified topics which are promising as the next big thing in science. The rest of the paper is structured as follows. Section 2 describes the data, Section 3 describes the problem, Section 4 presents the experimental results and Section 5 provides discussion.

## 2 Data Description

One could say the main element of science is an idea, invention or finding which occurs at the beginning of a scientific process. What follows is a period of scientific investigation, testing the idea in different contexts, proving the invention is useful or applying the findings in different scenarios. If proven to be valuable, new products or research is developed based on it. In our research, we rely on the fact that scientists are typically strict and consistent with naming conventions, enabling us to track the evolution of particular scientific topics through time.

In our research we have used the titles of scientific articles to identify when a scientific topic first appears, how frequently it appears through time, and when it stops being used. There are many databases of scientific articles in the world, but only some are open and available for research. Today, the biggest open database of scientific articles is known as the "Microsoft Academic Graph" which was released for research use in 2016. The database size is 104 Gigabytes, and it includes references to 125 million scientific articles from the year 1800 to 2015 from all areas of science. Each scientific article in the database is described by its: title, authors, their institutions, the journal or conference where it was published and the year of publication. The data is available from: <https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/>.



**Figure 1: Architecture of the system including NGram extraction, feature generation and machine learning. The generated model is used for classification to predict the popular topics as well as to identify the most important features.**

From the 125 million article titles, we extracted 2.5 million candidate topics, each corresponding to a phrase consisting of 1 to 5 consecutive words (also referred to as NGrams). The NGram must appear at least 100 times in the database of paper titles to be considered as a topic. Table 1 shows the distribution of NGrams

NGram	Total
1-Gram	300,000
2-Gram	1,000,000
3-Gram	800,000
4-Gram	300,000
5-Gram	100,000
All NGrams	2,500,000

**Table 1: The number of NGrams generated from the publication titles**

Figure 1 illustrates the process of feature generation and machine learning on the examples which represent the selected topics. The NGrams are generated from the paper titles, keeping only the frequent NGrams. For each frequent NGram, a feature vector is constructed using affiliations, conferences and journals of the papers in whose titles the NGram occurred.

For each topic, we find the longest span of years in which the topic appears in an article title at least once. Only topics which have the span of 15 years or longer are considered. This leaves us with about 1 million topics. Each topic is represented by a set of features describing the last 10 years before it became popular. The features include bag of affiliations, bag of journals and bag of conferences

of the publications in which the topic occurred. For each topic we report the total frequency over the 10 years and the slope of a line through the (year, frequency) points.

For instance, “SVM” as a topic has occurred in papers published by authors affiliated with Oregon State University (slope 0.5), Max Planck Society (slope 3), University of Waterloo (slope -0.5). We can see that the popularity of the topic “SVM” in the Max Planck Society has increased within the observed 10 years.

Each topic is described by approximately 55,000 features (23,000 journals, 1,300 conferences and 30,700 affiliations). Each topic is classified as either positive, if it became popular within the span of 15 years or as negative otherwise. Popularity is defined as a large difference in slopes of topic frequency in the 10 consecutive years compared to the following 5 consecutive years. We performed experiments varying the threshold (slope difference) from 1 to 5. A slope difference of 1 in our data results in 34% of examples being labeled as positive while a slope difference of 5 results in 20% of our examples being labeled as positive.

### 3 Problem Description and Algorithm

The problem we are solving is predicting early indicators suggesting which scientific topics are likely to become important in the future. The core task is to use the data from over 200 years of scientific discoveries from publications and to extract the early signs of a scientific topic becoming popular. Using machine learning algorithms, we have trained a statistical model to classify scientific topics into two categories: those which became important and those which did not. The model was trained on the data from



the year 1800 to 2015 to predict which topics will become relevant in the next 5 years from 2015.

For machine learning we used the Perceptron MaxMargin algorithm [2], an improved version of the perceptron algorithm. The improvement is in using two different margins, one for each class:

$$\text{MinPosMargin} = \frac{1}{\sqrt{\text{BadPosExs}}} \quad \text{MinNegMargin} = \frac{1}{\sqrt{\text{BadNegExs}}}$$

Where *BadPosExs* and *BadNegExs* are the numbers of misclassified positive and negative examples respectively in the previous epoch of training. In our experiments, we ran 3,000 epochs to build the model (meaning that we went through all the training examples 3,000 times). The learning rate was set to 0.02 in the case of no misclassifications in the previous epoch, and in the case of misclassifications, it was calculated as follows:

$$\text{LearningRate} = \frac{1}{\sqrt{\text{BadPosExs} + \text{BadNegExs}}}$$

As we are training a linear model, by examining the model itself, we can see the weights assigned to the features. The higher the weight, the more important the feature for the positive class. This means that by examining the model, we can see which affiliations, journals and conferences contribute the most to a topic becoming popular in the future.

#### 4 Experimental Results

We split the topics into a training (70%) and test set (30%), where the training set is used to train the model and testing set is used to test the model. The statistical model, trained with the MaxMargin Perceptron algorithm produced the following results on the testing data (see Table 2): Precision: 74.3 Recall: 71.7 F1: 73.0 for a slope difference of 1. This means the model correctly identifies 71.1% of the topics that became popular (recall) and 74.3% of the topics predicted to become popular really became popular (precision). As the slope difference increased the performance decreased, for instance, precision drops from 74.3 in slope difference 1 to 37.9 in slope difference 5. This is likely due to the increasing difficulty of the classification problem as the number of positive training examples decreases. The fact that the classification accuracy increases with the slope difference does not reflect improvement of the model's performance, as it is very close to the majority class (66% at slope difference 1, 80% at slope difference 5).

Slope Diff	Precision	Recall	F1	Accuracy
1	74.3125	71.6824	72.9737	63.1452
2	54.1432	60.3341	57.0712	60.1984
3	44.1246	46.7691	45.4084	69.2293
4	38.8584	47.1334	42.5978	76.6491
5	37.8595	45.1482	41.1838	82.9061

**Table 2: Precision, recall, F1 and accuracy on test data for slope difference from 1 to 5.**

Figure 2 shows the model's performance (estimated by a combination of precision and recall, F1) for 5 progressively stricter criteria of labelling topics as positive (slope difference 1-5).

We can see that the performance on the training and test set does not differ much on slope difference 1. As the slope difference increases, the performance on the test set drops relative to the performance on the training set.



**Figure 2: Graph of model performance (measured by F1, the higher the better) for test and train data and 5 slope differences.**

Looking at the resulting machine learning model we can see the following: if a scientific topic gets increasing attention from important research institutions (universities and research institutes), and is getting published by important journals and conferences within 10 years from its first appearance, then we can expect the increased use of the topic in scientific publications in the next 5 years.

In addition to the previous experiments, we have also built a perceptron model from scientific publications from 2006-2015. This model was used to predict future popular topics outside our dataset (5 years in the future from 2015). Looking at the results, one can notice several interesting topics predicted as promising. For instance: *proton proton collisions*, *higgs boson*, *quark*, *hadron*, *mobile augmented reality*, *variable quantum*, *molecular dynamics simulations*, *hadronic final states*, *search for dark matter*.

If we take a closer look at feature vectors of the promising topics during 2006-2015, we can notice for example that “*search for dark matter*” occurs in 56 papers with affiliation to *Purdue University* with a growing number of publications over the years (slope 4.14).

Another example is “*proton proton collisions*” which occurs in

- 610 papers with affiliation to the *Universite catholique de Louvain* with a growing number of publications over the years (slope 56.5).
- 8674 papers with affiliation to *CERN* with a growing number of publications over the years (slope 295.9).

Looking at the perceptron model trained on the data from 2006-2015, we can notice some of the most influential affiliations, conferences and journals are: *CERN*, *Journal of Proteomics & Bioinformatics*, *Industrial Research Limited*, *Circulation-*

*cardiovascular Imaging, Molecular BioSystems, Metamaterials, Atw-international Journal for Nuclear Power, Data Science Journal, IEEE Geoscience and Remote Sensing Letters, Columbia college, Princeton university school of engineering and applied science.*

## 5 Discussion

We analyzed 125 million articles from the “Microsoft Academic Graph” from over 200 years of scientific publications. In order to perform the experiments, we implemented the data preprocessing, feature generation and perceptron algorithm in C++. The resulting model was tested on a random 70/30 train/test split. The results show good performance, achieving F1 73.0%. The model predicts 71.7% of the scientific topics which became important in the history of science.

The possible direction for future work includes repeating the experiments on the new updated dataset, possibly considering the paper abstracts which have been made available in the dataset to be added to our feature set. It might also be beneficial to use the citation graph structure provided in the updated dataset. Another direction of future work would be applying the proposed approach to other similar datasets such as AMiner [3] or the Open Academic Graph [4, 5]. Yet another interesting direction of research would be to compare the performances of different machine learning algorithms and different data representations. Lastly, a more in-depth analysis of the topics predicted to become popular in the future would also be interesting.

We would also like to investigate ways to provide a publicly accessible online version of the system.

## ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and DataBench European Union Horizon 2020 project under grant agreement H2020-ICT-780966.

## REFERENCES

- [1] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo - June (Paul) Hsu, and Kuansan Wang. (2015) *An Overview of Microsoft Academic Service(MA) and Applications*. In *Proceedings of the 24th International Conference on World Wide Web(WWW '15 Companion)*. ACM, New York, NY, USA, 243-246.
- [2] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola. (2002) The perception algorithm with uneven margins. In *Proceedings of ICML 2002*, pages 379-386.
- [3] AMiner (Accessed Sept 2019) <https://aminer.org/>
- [4] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. (2008) ArnetMiner: Extraction and Mining of Academic Social Networks. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008)*. pp.990-998.
- [5] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. (2015) *An Overview of Microsoft Academic Service (MAS) and Applications*. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. ACM, New York, NY, USA, 243-246.
- [6] Billah, Syed Masum & Gauch, Susan. (2015). *Social Network Analysis for Predicting Emerging Researchers*. 27-35. 10.5220/0005593500270035.
- [7] Small, H., Boyack, K. W., and Klavans, R. (2014). Identifying emerging topics in science and technology. *Research Policy*, 48(8):1450–1467 – Elsevier.
- [8] Ohniwa, R., Hibino, A., & Takeyasu, K. (2010). Trends in research foci in life science fields over the last 30 years monitored by emerging topics. *Scientometrics*, 85(1), 111-127.



# Deep Language Classification for Relabeling of Financial News and its application in Stock Price Forecasting

Giulio Trichilo  
École Polytechnique Fédérale de Lausanne  
In association with The Jožef Stefan Institute  
giulio.trichilo@gmail.com

Miha Torkar  
Jožef Stefan Institute  
Jožef Stefan International Postgraduate School  
miha.torkar@ijs.si

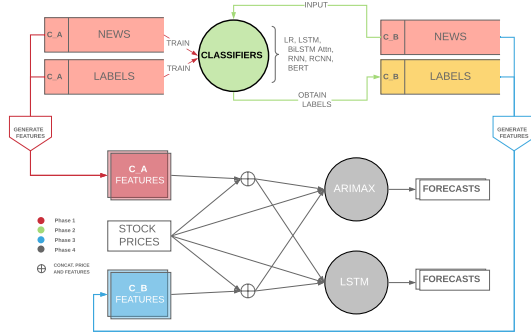


Figure 1: Workflow of the employed methodology.

## ABSTRACT

This paper aims at assessing the performance of the transfer learning task consisting of training set of classifiers on high frequency financial news data for 74 publicly traded companies, with domain specific labels. This source of data is provided by the Jožef Stefan Institute and is used exclusively for the purposes of this research. The trained classifiers are then used to attribute labels to an unlabelled source of high frequency aggregated news, *Event-Registry*. The aim is for the relabelled data to be used in the generation of exogenous features for use in time series forecasting of the companies' prices. It is found that using a fine-tuned BERT [1] model yields the most semantically coherent labels, and the features generated from the newly labelled data prove to yield the highest accuracy forecasts on held out price data.

## Keywords

Deep Learning, NLP, Language Model, Finance, BERT, Stocks, Forecast

## 1. INTRODUCTION

In recent years both natural language processing and algorithmic feature and signal based trading have been subject to an increasing level of automation, with statistical methods arguably being at the core of both. While the methods developed in both fields are still largely disjoint, in both these fields there is an attempt at modelling sequential data generating processes, be it natural language or price signals. Furthermore, empirical evidence strongly suggests that the dissemination of news regarding a financial entity such as a publicly traded company will in some way or another affect how active market participants will react.

This paper begins by characterizing the data from both corpora: the domain-specific labelled corpus and the *EventRegistry* corpus will hereby be referred to as  $C_A$  and  $C_B$  respectively. This data is initially used for the training of word2vec [7], doc2vec [5] and fasttext [2] to generate word embeddings to be used downstream in a set of deep language classifiers: LSTM, BiLSTM with Attention [6], CNN [3], and RCNN [4]; a logistic regression model serves as a baseline. Furthermore, BERT is employed, however as is standard practice, pretrained word vectors constitute the model's initial state which is then fine tuned on the  $C_A$ . The trained classifiers are then used to attribute labels from  $C_A$  to  $C_B$ . While there is no standard metric for the evaluation of the pertinence of the attributed labels, this is approached through employment of semantic similarity metrics and t-SNE in an attempt to extrapolate a relationship between semantic and projected spatial clustering.

The time series forecast setting consists of two separate sub-tasks performed twice, once on the data from  $C_A$  where the true labels are available, hence the feature vector construction is not subject to potential mislabelling in terms of semantic incoherence, then again on the relabelled  $C_B$  corpus with BERT labels as basis for feature construction.

Each set of features is used firstly as a series of exogenous regressors in an ARIMAX setting, this primarily in order to gauge coefficient significance and quality of the forecast with respect to the baseline of no exogenous regressors. Concurrently, each set of features is used as inputs to a two layer LSTM followed by a feed-forward net, which allows forecasting of both the stock price and the news series, however it is to be kept in mind that the explicit exogeneity relationship which characterizes ARIMAX is not maintained in the LSTM setting.

## 2. DESCRIPTION OF THE CORPORA

Corpus  $C_A$  consists of 3M timed news headlines between Jan 1st 2006 and Dec 31st 2018. For each headline, the associated label as well as the company in question are given. There are 53 labels however the distribution of labels over the news entries is heterogeneous, resulting in an imbalanced dataset. Therefore a balanced subset of the 120,000 entries per label from the 20 most frequent labels (yielding 2.4M entries) was selected. Of these, the train, validation, test split was selected as 75/15/10. This data split is used in the training, validation and testing of all classifiers examined. The selected subset of data contains news entries from 74

	mean	min	25%	50%	75%	max
$C_A$	4389	20	1298	2493	5550	37800
$C_B$	13489	43	2029	5522	14815	236856

**Table 1:** Distribution of news counts for the 74 companies.

publicly traded companies.

On the other hand,  $C_B$  consists of roughly 1M timed news headlines from Jan 1st 2014 to May 31st 2018, exclusively for the 74 companies examined. This is the corpus on which labelling is to be performed. In Table 1 summary statistics for  $C_B$  and for the subset of the  $C_A$  consisting only of the 74 companies examined is presented, both at their unadulterated frequency.  $C_A$  and  $C_B$  have a mean news headline length of 11.05, 11.36, with standard deviation of 4.52 and 9.34, respectively. Their empirical distributions are approximately  $\chi^2$ -distributed.

Finally, a second corpus from *EventRegistry* consisting of 50 *dmoz* labels was made available, however this corpus contains no associated company information. From this corpus, only those headlines whose class belongs to a subset of 20 top level categories, chosen by hand due to similarity with  $C_A$ 's labels, has been kept. This data subset (hereby corpus  $C_{B2}$ ) is not used in modeling and plays only a very minor role in the evaluation of the relabelling performance in section 3.3.

### 3. CLASSIFICATION AND LABELLING

All classifiers are trained on  $C_A$  according to the chosen split. This section begins by outlining the methods used for word embedding generation used in all classifiers except BERT, then covers overall model performance. In order to assess the ability for a given classifier to attribute semantically consistent labels to  $C_B$  corpus, cosine similarity between the label vector and its neighborhood, defined here as the subset of the 30 words with the highest empirical probability of occurring for each label, according to each classifier, is computed.

#### 3.1 Generation of Word Embeddings

In standard literature, in order to perform text classification the elements of a labelled corpus  $C = \{(c, D)\}$ , where  $(c, D)$  is a class-document pair, the elements  $w \in D$ , where  $D \subset V$  and  $V$  is the vocabulary, must be mapped to a vector space, typically  $\mathbb{R}^n$ , where  $n = \{|V|, \mathbf{d}\}$  depending on whether a count based model is used or whether one aims to represent each word as a (typically dense)  $\mathbf{d}$ -dimensional vector.

In general, a neural embeddings model aims at finding

$$\hat{\theta}, \hat{\mathbf{E}} = \operatorname{argmin}_{\theta, \mathbf{E}} L$$

Where  $\mathbf{E} \in \mathbb{R}^{|V| \times \mathbf{d}}$  is the embeddings matrix which can be then passed on to downstream tasks such as text classification, and  $L$  is a loss function over the corpus, the context for each word in the corpus, given the embeddings matrix, and all other trainable parameters  $\theta$ .

In this paper, word2vec, doc2vec and fasttext<sup>1</sup>, using Con-

<sup>1</sup>No subword information was used as no significant accuracy was

textual Bag of Words, Distributed Memory (DM), and Bag of Tricks respectively, are used to obtain three separate embeddings matrices given the training corpus. The embeddings are chosen to have  $\mathbf{d} = 300$ . All models were trained for 20 epochs, with a minimum count of 4, and a context window of size 7. All other parameterizations are as in [7], [5], [2], respectively.

#### 3.2 Classifier performance on $C_A$

In this section performance of the LSTM, BiLSTM with Attention, CNN, and RCNN, and BERT, is analyzed. Results of training a logistic regression model serve as a basis for comparison.

##### 3.2.1 Logistic Regression

In order to gauge classifier performance all generated word embeddings are used in training a logistic (softmax) regression classifier, as this is taken to be the simplest model trainable on the data<sup>2</sup>. This classifier aims at maximizing

$$P(c | D) = \operatorname{softmax} \left( W_c \sum_{w_i \in D} \operatorname{embed}_{\mathbf{E}}(w_i) \right)$$

where the summation term yields the embedding for the document<sup>3</sup>. The classifier is trained on all three sets of word embeddings, with 72% average class accuracy for fasttext, 69% for word2vec and 68% for doc2vec. The labels attributed to misclassified samples for each class are generally evenly distributed amongst the other 19 classes.

##### 3.2.2 Deep Word Embedding Classifiers

LSTM, BiLSTM with Attention, CNN, and RCNN are the four deep classifiers tested. As fasttext embeddings have yielded the highest accuracy on  $C_A$ , these will be the embeddings used for these models. This choice does not in general guarantee classifier optimality, however it gives grounds for standardized comparison. In order to further enforce this, all LSTM-based models were trained with the following common hyperparameters:

$ V $	$\mathbf{d}$	LSTM_out	batch_size	epochs
263,088	300	256	64	5

For all LSTM-Based models the initial hidden and cell states were set as  $(h_0, c_0) = (\mathbf{0}, \mathbf{0})$ . For the CNN the following hyperparameters were given. The model was trained for 5 epochs with the same embeddings as the previous cases. Furthermore, one channel was used in input and eight in output. Kernel sizes were 2,3,4, the stride was set to 2 for all layers and the vertical padding to 1.

All models were trained using Cross Entropy as the loss function and ADAM as the optimizer, with a learning rate  $\eta = 10^{-3}$ , no weight decay, and numerical stability parameter  $\varepsilon = 10^{-8}$ .

gained in subsequent use of the embeddings.

<sup>2</sup>Logistic Regression with Bag of Words as input, trained on a subset of data exclusively from the year 2017, yields an average class accuracy of 70%

<sup>3</sup>Obtaining the document embedding from the word embeddings is not a trivial problem, however addition is sufficient for the purposes of this classifier.

### 3.2.3 BERT

BERT leverages masked language modeling and the encoder from the transformer architecture in order to learn contextually coherent word representations. Unlike the previous cases BERT is initialized with its own pre-trained embeddings; all hyperparameters are kept as in BERT-Base as specified in [1]. The model was trained for 5 epochs.

Given that BERT uses wordpiece for tokenization, the size of its pretrained vocabulary is not indicative of the true dimensionality of vocabulary space. The model was adapted for classification trained using Cross Entropy as the loss function and ADAMW as the optimizer, with a learning rate of  $\eta = 10^{-3}$ . Furthermore a scheduler with a linear warmup is implemented, with 100 warmup steps.

For all models, a weighted average of precision and recall, along with the F1 scores of the best and worst scoring classes are given in Table 2.

### 3.3 Evaluation of Labelling on $C_B$

In order to attempt at quantifying the pertinence of the domain-specific labels attributed to  $C_B$ , the cosine similarity between the label and the 30 most frequent words attributed to it (net of english stopwords and special characters), constituting a threshold on the empirical distribution of words for each label, is computed for all classifiers; then, the empirical similarity quartiles are computed for said classes, and the maximum over all classes for each quartile is reported<sup>4</sup>. In order to have some idea of how this compares to labelled data, this is repeated both for  $C_A$  and for  $C_{B2}$ . The results are reported in Table 3<sup>5</sup>.

In accordance with intuition, those labels with worse test performance across models have a less relevant set of top words associated to them. It is interesting to note how the similarity between BERT’s attribution of  $C_A$ ’s labels on  $C_B$  is in all cases higher, and the standard deviation lower, than is the case with  $C_{B2}$ . It is to be noted that these are not fair grounds for comparison as the corpora are different, however this does point to BERT’s ability to capture semantic similarity in a more ‘natural’ manner than the other models.

## 4. FEATURE GENERATION FROM NEWS

Feature vectors are constructed by taking the relevant news events for each company for all trading days between Jan 1st 2014 to Dec 31st 2017. For each trading day, for each company, the count of the events for each category is assigned as the elements of the feature vectors (20 dimensional). The labels are the original ones for  $C_A$ , and  $C_A$ ’s BERT-attributed labels for  $C_B$ . The price series data used is the daily close price adjusted for dividends. The following operations were performed in order to assure consistency in the construction of feature vectors, for each company:

- For each day, obtain the feature vectors as described above for three time intervals: Pre-Hours (00:00-09:30), During

<sup>4</sup>The maximum is taken as the relabelled dataset is in all cases unbalanced.

<sup>5</sup>In addition, t-SNE is used to project label and neighborhood into  $\mathbb{R}^2$ ; observable clusters are, expectedly, less well defined on the relabelled  $C_B$  than the clusters identifiable when projecting  $C_A$ .

Trading Hours (09:30-16:00) and After Hours (16:00-24:00). Any day over the entire year (365 days) where no events happen is attributed a zero vector.<sup>6</sup>

- Given the adjusted close price is being used, the assumption is made that today’s close will be affected by news from today’s pre-trading hours, today during trading, as well as yesterday’s after hours. Therefore yesterday’s after hours vector is added to today’s pre-trading hours vector and to today’s trading hours vector.
- Given that the trading days a year are 252, feature vectors indexed at a non trading day are made to contribute to the next trading day (ex: the resulting feature vectors for a weekend are added with next monday’s).

This construction assures the removal of any look-ahead bias (we are only interested in the scenario where the news affects the price, and not when the news event manifests itself as a reaction to a change in the stock price), however this construction does assume that news on a given day takes at most one trading day to incorporate into price.

## 5. FORECASTING USING NEWS

In this section the predictive performance for feature vectors generated from both  $C_A$  and the  $C_B$  with BERT-attributed  $C_A$  labels will be evaluated. The training period is the first three trading years: Jan 1st 2014 - Dec 31st 2016, and the held out period is the last 52 weeks.

### 5.1 Features as exogenous variables

An ARIMAX model is initially employed to test for significance of the categories of the events. In this setting, each dimension of the feature vectors constitutes a univariate time series. It is therefore these 20 exogenous series which are used as regressors in the ARIMAX setting.<sup>7</sup> For each price series the optimal order,  $ARIMA(p, d, q)$ , is computed based on SBIC, and the inferred order is maintained when including the respective exogenous variables<sup>8</sup>. It is found that  $3.68 \pm 2.07$  categories are statistically significant in predicting the price for  $C_A$ , and  $1.78 \pm 1.55$  for  $C_B$ .

### 5.2 Features as inputs in LSTM

A unidirectional two-layer LSTM network is employed in order to gauge performance of price as well as news forecasting. The inputs to the networks are, for each time step, the 10 previous observations for both the close price and the 20 news series. Minmax scaling is used in order to render the input space more isotropic and promote gradient stability; all variables are then rescaled after training.

In Table 4 error metrics are computed for the holdout period from Jan 1st 2016 to Dec 31st 2017 (the final year of data). The Diebold-Mariano test is computed pairwise for each forecast:  $C_A$ ,  $C_B$ , and the vanilla ARIMAX and LSTM

<sup>6</sup>This yields 22K, 40K, 49K events for  $C_A$ , and 218K, 270K, 367K events for  $C_B$ , for the respective brackets.

<sup>7</sup>The training period must for some stocks be lengthened to compute coefficient significance (guarantee exogenous nonsingularity).

<sup>8</sup>Inferred order directly including exogenous series would sometimes yield  $p = q = 0$ ,  $d = 1$ ; this is never the case on just the series.

Model	Embed	Wavg. Precision	Wavg. Recall	Best Class	F1	Worst Class	F1
LR	fasttext	72%	70%	Exploartion	1.00	Insider-Trading	0.35
LR	word2vec	72%	69%	Exploration	1.00	Insider-Trading	0.35
LR	doc2vec	73%	68%	Credit	1.00	Insider-Trading	0.28
LSTM	fasttext	74%	74%	Credit	1.00	Labor-Issues	0.46
BiLSTM	fasttext	71%	68%	Investor-Relations	0.85	Marketing	0.40
CNN	fasttext	75%	74%	Credit	1.00	Analyst-Ratings	0.41
RCNN	fasttext	75%	73%	Exploration	0.99	Insider-Trading	0.44
BERT	BERT	79%	78%	Legal	1.00	Stock-Prices	0.52

**Table 2:** Model Performance on  $C_A$ 's Test Set.

	LR	CNN	RCNN	LSTM	BiLSTM	BERT	$C_A$	$C_{B2}$
mean	0.142	0.103	0.118	0.119	0.148	0.388	0.507	0.281
std	0.282	0.274	0.285	0.274	0.272	0.294	0.381	0.334
min	-0.219	-0.213	-0.213	-0.213	-0.213	-0.107	-0.054	-0.150
25%	-0.021	0.009	-0.021	-0.017	0.067	0.176	0.365	0.143
50%	0.165	0.126	0.155	0.107	0.145	0.395	0.579	0.282
75%	0.316	0.269	0.294	0.301	0.279	0.610	0.762	0.435
max	0.666	0.667	0.668	0.666	0.745	0.802	1.000	1.000

**Table 3:** Maximum cosine similarity quartiles across all classes for all models on  $C_B$ . The last two columns act as a baseline showing similarity scores for the two labelled corpora.

	mae	rmse	minmax	D.M.		
<i>ARMIAX</i>						
NONE	4.916	5.898	0.063	-	24	25
$C_A$	4.399	5.614	0.055	35	-	44
$C_B$	4.376	5.482	0.061	31	47	-
<i>LSTM</i>						
NONE	7.158	8.033	0.103	-	20	31
$C_A$	1.553	1.930	0.018	22	-	41
$C_B$	1.001	1.348	0.015	50	54	-

**Table 4:** Median forecast error metrics across all stock prices and forecast disparity counts between models (number of stocks for which a given forecast prevailed).

forecasts respectively.<sup>9</sup>

It is found that when no news is used the model is more likely to learn a degenerate prediction (a constant) than when news is used as input. However, forecasts using news are for all nondegenerate cases more volatile than those without. Since this behavior appears to be pseudo-deterministic, degenerate predictions were left in when calculating error metrics and performing the DM test.

## 6. CONCLUSIONS

In the present work it has been shown that BERT is able to perform the classification task with the highest accuracy out of all models, as well as yield the most semantically

<sup>9</sup>While the test does assume the loss differential to be covariance stationary, which isn't often the case for ARIMAX, plotting all three sets forecasts for this model class seems to empirically validate the verdict of the test statistic (in cases when  $DM \sim \mathcal{N}(0, 1) \gtrless \pm 1.96$ ).

consistent labels on the previously unseen corpus  $C_B$ . Furthermore, it has been shown that utilizing features generated from news for forecasting stock prices for the given sample of companies over the selected interval yields significantly better predictions than not using news for ARIMAX. The LSTM network however seems to predict prices with much higher accuracy in all nondegenerate cases, with news features from  $C_B$  yielding the set of predictions with lowest median error across all measures, indirectly pointing to BERT's efficacy in relabelling. In terms of news forecasts with this model however, it is with  $C_A$ 's data that news series forecasts are on average more reliable.

## 7. ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 675044.

The first author is grateful to Dunja Mladenec and the E3 department for the opportunity for a summer at the Jožef Stefan Institute.

## 8. REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018.
- [2] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *EACL*, 2016.
- [3] Y. Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 08 2014.
- [4] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, 2015.
- [5] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *ArXiv*, abs/1405.4053, 2014.
- [6] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *ArXiv*, abs/1703.03130, 2017.
- [7] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

# Semantic Enrichment and Analysis of Legal Domain Documents

M. Beshar Massri  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
beshar.massri@ijs.si

Erik Novak  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
erik.novak@ijs.si

Sara Brezec  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
sara.brezec@ijs.si

Klemen Kenda  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
klemen.kenda@ijs.si

## ABSTRACT

In text mining document enrichment processes are used to improve information retrieval. Document enrichment helps us extract metadata from the text which can then be used in document classification.

This paper presents the legal domain document enrichment process and analysis of the enriched data. The process of enriching the documents with multiple layers of annotations is described. The focus is on legal domain documents data set, but the proposed procedure can be generalized to any type of documents.

## Keywords

document enrichment, semantic annotations, ontology, analysis, legal domain

## 1. INTRODUCTION

Document enrichment process helps to improve information retrieval. Nowadays, more and more data has to be processed which makes information retrieval systems extremely valuable. Using document enrichment, more information can be gained about the documents which can be optimized for retrieval.

In the legal domain, extracting meta data about the legal domain documents improves building search engines which are designed to help lawyers efficiently access documents related to a certain topic. In this paper, we present an enrichment process of the legal domain documents. Different types of annotations are used to enrich the data; word-level features which are associated with word information, Wikipedia concepts gained by the process of Wikification and InforMEA ontology terms that cover the field of Environmental Law and Governance. Next, preliminary analysis on the enriched documents is used to review the results. Throughout the paper the focus is on legal domain documents. This approach can be generalized to other document data sets. Our contribution is applying semantic annotation and mapping with ontology on environmental legal domain documents.

The remainder of the paper is structured as follows: Sec-

tion 2 is related work. Next, the data set is described in section 3. Section 4 presents the methodology used for the document enrichment process. Analysis of the results is in section 5 and finally, we present future work and conclusion in section 6.

## 2. RELATED WORK

Much work has been done on semantic enrichment of text. Some tools provide a generic pipeline that can be applied and embedded into more complex pipelines. Such pipelines include word and sentence tokenization, part of speech tagging, dependency parsing, and named entity recognition. Examples of such tools are software packages or libraries for different languages, like Spacy [5], Scikit Learn [14], Stanford CoreNLP [11], and MITIE [4].

Semantic enrichment methods have been used to improve the features when building classification models of documents in different domains. An example of this can be found in [7], where two levels of semantic enrichment were used before and after training to classify medical domain documents. In [1], they used dependency parsing, ProbBank [9], and hypernyms from WordNet [13] among other syntactic and semantic features to build relation classification models for the SemEval-2010 Task 8. We also see in [10], the use of mapped cross-domain ontologies in improving information retrieval in the biomedical and chemical domain documents.

In this paper, some of the tools and techniques will be used plus others, mentioned above, and applied to the legal environmental domain documents, providing further analysis about information extracted from the corpus based on the enrichment process.

## 3. DESCRIPTION OF DATA

We used EUR-Lex, an online service that provides different documents regarding the European Union, as a source to extract our data [3]. For each document, a set of descriptors or keywords was provided among other metadata, in addition to the document title and text. Based on the descriptors and the language of the text, the environmental legal documents were filtered which were provided in the English language

and used as the main source of data for document enrichment. The resulting data set, after filtering and cleaning, was around 72k documents.

After preliminary inspection of the data, the documents vary greatly in length. The longest document contains about 560k words whereas the shortest contains 27 words. Nevertheless, approximately 99% of the documents have less than 30k words, 90% of them have less than 5k words and 66.6% have under a 1000 words. Sometimes it can be noticed that classification models produce better results on sets of documents with similar length. Mentioned numbers indicate the potential of providing more precise classification on a set of documents where only few documents are removed from the initial data set.

## 4. DATA ENRICHMENT PROCESS

### 4.1 Standard NLP pipeline Annotations

As a first step in data enrichment process, the traditional natural language processing analysis methods were used. The Stanford CoreNLP library was chosen, which is a set of human-language technology tools developed at Stanford University [11]. Using the library, the documents were tokenized into words and then a set of basic syntactic and semantic information was extracted for each word:

- The tokenized word
- The lemma, or dictionary form of the word
- The part of speech of the word in the text.
- Set of synonyms for the word using WordNet lexical database [13], when applicable.

In addition, entity recognition methods were used to identify entities that were categorized into following 11 category classes:

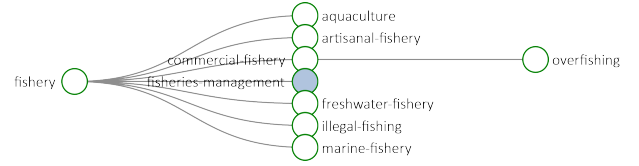
- Named entity classes: PERSON, LOCATION, ORGANIZATION, and MISC
- Numerical entity classes: MONEY, NUMBER, ORDINAL, and PERCENT.
- Temporal entity classes: DATE, TIME, and DURATION.

The MISC category represents an entity mention that was not classified in any of the mentioned classes. An example of these entities are document types ('Regulation') and languages ('English'). Other classes are self-explanatory.

### 4.2 Wikification

The second annotation step was wikification, which is extracting entities with a relevant Wikipedia concept from the text. The JSI Wikifier tool was used, which is a service developed in Jozef Stefan Institute, that annotates a given raw text with annotations each representing a Wikipedia concept [8].

For each document in our data set, we used Wikifier on the raw text provided and obtained a list of annotation objects; each contains the following information:



**Figure 1: A snapshot that contains a subset of the InforMEA ontology tree.**

- The annotation name representing the Wikipedia concept
- Wikipedia page URL of the annotation
- Wiki data classes: the set of classes from WikiData knowledge base [6] that this annotation belongs to.
- One of the DBpedia [1] identifiers that corresponds to the annotation.
- The page rank score of the annotation.
- The cosine similarity between the the document text and the Wikipedia page that the annotation represents.

### 4.3 InforMEA Ontology

Finally, to provide information about the potential environmental categories that the documents are categorized into, InforMEA ontology was used to map the document with relevant environmental ontology terms. The ontology has 532 unique terms that form a hierarchical structure based on the 'broader' relation between ontology concepts. A subset of the ontology tree visualization representing the branch 'fishery' is shown in figure 1. More detail, along with the ontology tree, is available on GitHub [12].

To annotate the documents with InforMEA Ontology terms, a simple string matching method was used between the ontology terms and the metadata provided. For each document, the following enrichment data was used to search through for words that matched with any ontology terms:

- The normalized words of the documents
- The synonyms of those words
- The wiki-data classes of the Wikipedia annotations extracted from the document

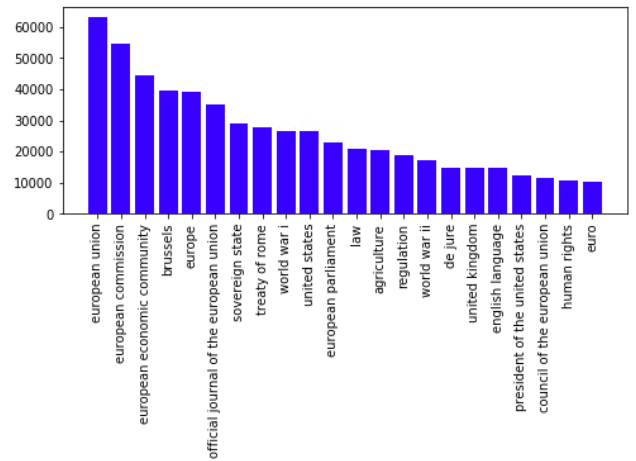
The reason for using the wiki-data classes instead of the Wikipedia concepts themselves is that the Wikipedia concepts are usually too specific to match with an ontology term, whereas the Wiki data classes represent the topic or the category that this concepts falls into. In fact, the Wikipedia concepts were included in the initial experiments, but had to be omitted later as they did not produce any matches.

## 5. ANALYSIS OF RESULTS

After annotation was done, extracted information was analysed to get an initial evaluation about the nature of the corpus.

Out of the 72k documents in the corpus, 157k unique Wikipedia concepts were extracted, with only 22 of them having occurrences in over 10k documents. Furthermore, about 50k concepts appear in only one document and 100k concepts appear in up to three documents. This indicates that most of the concepts are unique to the documents. In regards to Wikipedia concepts, the most frequent Wikipedia concepts are shown in Figure 3. The majority of the concepts can be associated with the European union. From the same figure, some concepts can be associated with law and environment, such as “law”, “agriculture” and “regulation”. This indicates that the process of wikification is able to acquire relevant information. In addition, Geo-spatial concepts are extracted through the process. Their presence can be acknowledged in the country names which are also amongst the most frequently found Wikipedia concepts. Nonetheless, the wikification process was able to find concepts for which connection with the documents is not clear. This will be investigated in future work.

Most frequently occurred LOCATION named entities were country names. In the ORGANIZATION class legal bodies were mainly found; almost all of them were associated with the European Union. In almost every document at least one ORGANIZATION and one LOCATION entity appeared.



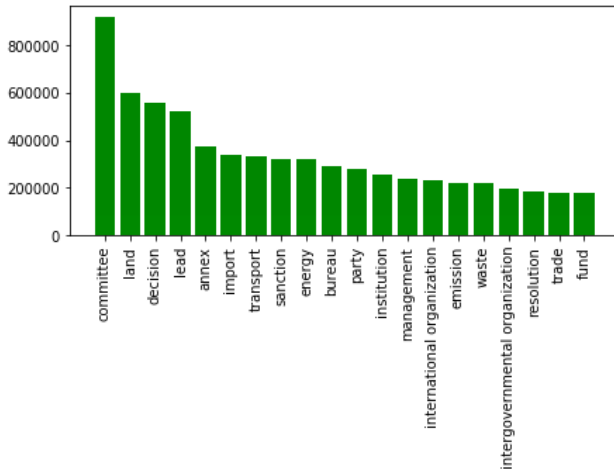
Entity Type	Count (approx. $\times 10^7$ )
ORGANIZATION	0.28
NUMBER	0.95
PERSON	0.08
DATE	0.16
MISC	0.14
ORDINAL	0.02
PERCENT	0.04
LOCATION	0.14
MONEY	0.01
TIME	0.01
DURATION	0.03

In comparison to Wikipedia concepts and entity results, a similar pattern of results was obtained from the analysis performed on the word-level features. Therefore, we omit the representation of the outcomes.

The most frequent ontology term ‘committee’ can be found in other annotation classes as ‘commission’. Additionally, ontology terms associated with organizations, logistics and the environment appear amongst the most frequent ontology terms.

In conclusion, a semantic enrichment methodology consisting of three main processes; annotation, wikification and





**Figure 5: The 20 most frequent ontology terms. Terms are chosen from the aggregated set of normalized words, word synonyms, and wikidata classes of the extracted wikipedia annotations.**

mapping to the InforMEA ontology, was performed on legal domain documents. In addition, the analysis on the extracted metadata was provided on the corpus scale to examine the nature of the dataset semantics.

Based on the analysis, some problems were observed with the wikification process as it produced a few unrelated matches. The plan is to address this problem in more detail, observe the reasons behind them, and if possible, try to partly solve the problem.

Regarding the named entities annotation, consideration of adding more finely-tuned annotations, like geo-spatial locations, would help in providing more accurate metadata about the documents. Furthermore, improvement could be made on the baseline string matching that was used to match documents with InforMEA ontology terms. By building classification models, the intention is to use the extracted annotations as features among others.

Finally, the enrichment was mainly done to provide additional metadata on the documents that will be used in later processes. Later plans for further work will be to use the annotations in query expansion to improve legal document retrieval.

## 7. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and EnviroLens European Unions Horizon 2020 project under grant agreement No 821918 [2].

## 8. REFERENCES

- [1] DBpedia knowledge graph. <https://wiki.dbpedia.org/>. Accessed in: August 2019.
- [2] EnviroLens project. <https://envirolens.eu/>. Accessed in: August 2019.
- [3] Eur-Lex. <https://eur-lex.europa.eu/homepage.html>. Accessed in: August 2019.
- [4] MITIE: Mit information extraction. <https://github.com/mit-nlp/MITIE>. Accessed in: August 2019.
- [5] spaCy industrial-strength natural language processing in python. <https://spacy.io/>. Accessed in: August 2019.
- [6] WikiData the free knowledge base. <https://www.wikidata.org>. Accessed in: August 2019.
- [7] ALBITAR, S., ESPINASSE, B., AND FOURNIER, S. Semantic enrichments in text supervised classification: Application to medical domain. In *FLAIRS Conference* (2014).
- [8] BRANK, J., LEBAN, G., AND GROBELNIK, M. Annotating documents with relevant wikipedia concepts.
- [9] KINGSBURY, P., AND PALMER, M. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)* (Las Palmas, Canary Islands - Spain, May 2002), European Language Resources Association (ELRA).
- [10] KÖHNCKE, B., AND BALKE, W.-T. Enriching documents with context terms from cross-domain ontologies. *Information and Media Technologies* 10, 2 (2015), 294–304.
- [11] MANNING, C. D., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S. J., AND MCCLOSKEY, D. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations* (2014), pp. 55–60.
- [12] MASSRI, M. Ontology tree visualizer. <https://github.com/besher-massri/OntologyTreeVisualizer>. Accessed in: August 2019.
- [13] MILLER, G. A. Wordnet: A lexical database for english. *Commun. ACM* 38, 11 (Nov. 1995), 39–41.
- [14] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.



# Health News Bias and its impact in Public Health

J. Pita Costa, F. Fuat,  
L. Stopar, M. Grobelnik, D. Mladenčić  
Quintelligence,  
Jozef Stefan Institute, Slovenia Slovenia

A. Košmerlj, E.  
Belayeva, L. Rei  
Jozef Stefan Institute,  
Slovenia

G. Leban  
Event Registry,  
Quintelligence,  
Slovenia

S. Fischhaber  
Analytics Engines,  
UK

J. Wallace  
Ulster University,  
UK

## ABSTRACT

The impact of health-related news in today's society is increasing as is the awareness of the globalization of the worlds' habits and threats, and the impact on the continuous pursuit of a better quality of life. The risk of news media bias and the consequences it might have in the population is of great concern for public health, as are the available resources to identify the bias and further explore the news stories. In this paper we discuss several aspects, angles and perspectives on news media bias in the health domain, with a particular focus on digital epidemiology. We also present decision support tools developed to support decision makers in these explorations in the context of the MIDAS project, leveraging Big Data analytics to support decision making in public health. The presented resources provide health professionals with a global perspective on the worldwide news coverage of monitored health topics (such as, e.g., infectious diseases, mental health or childhood obesity), together with a workflow of tools allowing them to explore potential bias. Moreover, we discuss the specific challenges of news bias in the health domain, analyzing some typical examples, and using the Event Registry technology to further explore them. The exploration potential of the latter, in the health domain, is enhanced with the integration of an automated classifier based on MeSH Headings that allows researchers to explore the news using a similar workflow to that of exploring biomedical research in PubMed.

## CCS CONCEPTS

• Information Systems • Human-centred Computing • Life and Medical Science

## KEYWORDS

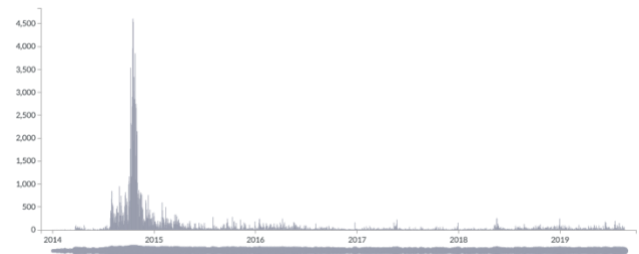
Data mining, health news, news media bias, Big Data, Public Health, digital epidemiology

## 1 HEALTH NEWS MEDIA BIAS

News media bias is a ubiquitous phenomenon that has generated various research studies in different fields. Practically any media outlet can be biased, but the public should be aware of it and news media bias should be minimized thereby offering more objectivity to the news reporting. Health related news, in particular, have a high impact on the population that tends to be more sensitive to their content. It is fairly well known that the media plays an influential role in public responses to health issues [6]. Although, the bias in health-related news can be considered in the same light as the overall news bias, with similar effect in most cases, it also

has very specific aspects deriving from the domain it is based on and the kinds of stakeholders it relates to. Often, the complexity of the information (due to the continuous innovation in medicine) along with the lack of detail can lead to misinterpretations and unconscious bias both at the media outlet and its audience. This is a common problem in the communication of science [13].

Examples of these are frequent in the context of precision medicine, where some difficult concepts and methods from genetics and life sciences play a key role while being a sensitive topic within the common public opinion.

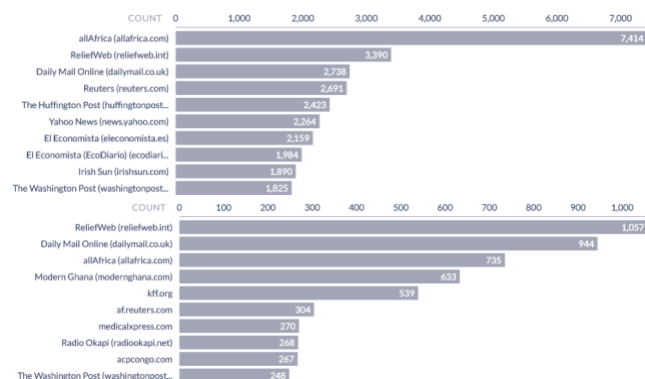


**Figure 1: MIDAS news dashboard screenshot showing a temporal intensity visualization module for the query *Ebola virus disease* to analyze and compare the media coverage on the disease outbreak during 2014 and 2019, considering only news sources located in the USA.**

Another angle on news media bias is the amount of news on certain disease-related aspects, that are more abundant in media sources that are located far from where they are occurring, providing us with an idea that the occurrence is local. An example of this in the public health scenario, is the large amount of news published by news sources located in the USA about the Ebola outbreak, and the small number of cases detected in this country. The frequency of news items can be sometimes confused with its potential impact in the local citizens, by the less informed audiences. The transfer of that unclear message to a diversity of social media channels is then inevitable, as well as the subsequent accelerated proliferation of the misinformation and unconscious news bias.

The chart in Figure 1 shows two perspectives on health news bias while representing the news on Ebola virus disease media coverage limited to news sources in the USA. On one hand the peak in 2014 is not representative of the low number of cases identified in the USA. On the other hand, the weight of the disease in 2015 and now in 2019 is not representative to the high relevance of this topic to the global public health today. In July 17, 2019, the World Health Organization (WHO) was once again announcing an Ebola Outbreak in Congo with public health emergency of international

concern [14]. Though, the news coverage this time is much more local than it was back in 2015 as the reader can see in Figure 2.



**Figure 2: The coverage of the Ebola outbreak in 2014 (above) and in 2019 (below), showcasing the very different top 10 news sources covering the similar event.**

A well-known generator of health news media bias was the case of the Google Flu Trends, a good example of collective intelligence estimating the influenza activity for more than 25 countries. This system was based on the queries for influenza related keywords on the Google search engine [3]. The Influenza season of 2012/2013 showed the inaccuracy of this system that, until then was closely following the data collected by the Centers for Disease Control (CDC), as seen in Figure 3. Although being more a case of algorithm bias per se, it was the responsible for false conclusions that could have had a bigger impact without the classical mechanisms in place by National and International Institutions.

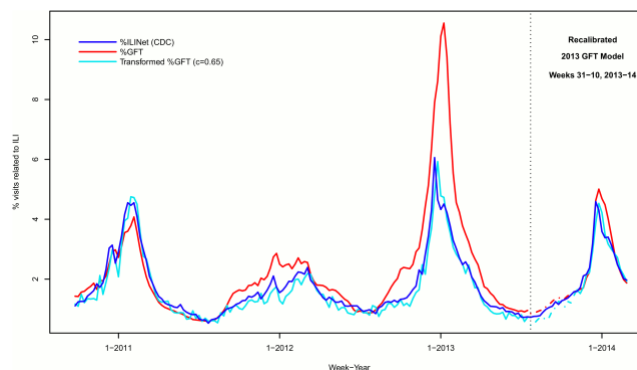
## 2 CHALLENGES OF HEALTH NEWS

Unlike most news topics, health related topics are often close to the interest and well-being of the general public, with a large impact in the social media [2]. An example of that is the ongoing worldwide public discussion about child vaccination. However, aside from the popular diseases (such as influenza, measles, etc.), the media coverage is often not complete or does not provide an accurate coverage (not all topics are *news worthy*). The rising popularity of a certain disease implies the increase of the references to it in the media, not in parallel with the status of the disease itself. This sometimes falls into what is known as *mainstream bias*, i.e., a tendency to report what everyone else is reporting, and to avoid stories that will fall out of the core of popular news.

On the other hand, the awareness of a certain disease or the general status of public health is not always well represented in the media. Most of the times this lack of representation reflects the incomplete awareness of the general public to the state of the health nationwide. It is also the case when that awareness is higher in some countries and smaller in others. This is often the case differentiating the so-called developed countries to the so-called 3rd world countries. An example of this is the coverage of the news about the Zika virus outbreak in 2015/16.

Another angle that is relevant to this discussion is the different concepts of news media bias and what is ‘news-worthy’ discussed in [5]. Although the acknowledged importance of a complete global coverage of the status of the health of the population, some aspects of health have higher priority than others, independently of their relevance in the Public Health context. These priorities are defined by the media houses and publishers according to the expectation on the impact that the news will have in their audiences. In a more extreme sense, sensationalism is the bias in favor of the exceptional over the ordinary, aiming to give the impression that rare events, such as a victim of the Ebola outbreak in the USA, are more frequent than common events, such as a child with Type 2 Diabetes originating from obesity and a sedentary lifestyle.

A more accurate analysis of the media coverage of an epidemiological phenomenon needs to be handled in the same way studies are. While an epidemiological study results may reflect the true effect of an exposure to the development of the outcome under investigation, it should always be considered that the findings may in fact be due to an alternative explanation [8].

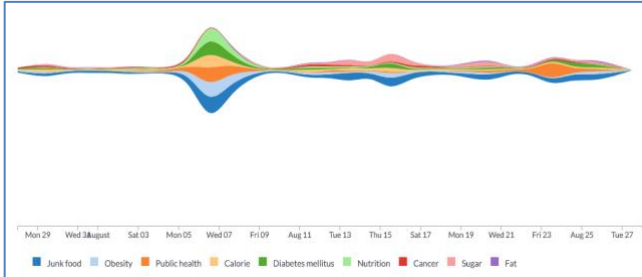


**Figure 3: Incorrect estimations of the Google Flu Trends (in red) based on online queries, against the CDC (in dark blue) for the influenza season of 2012/13 [10].**

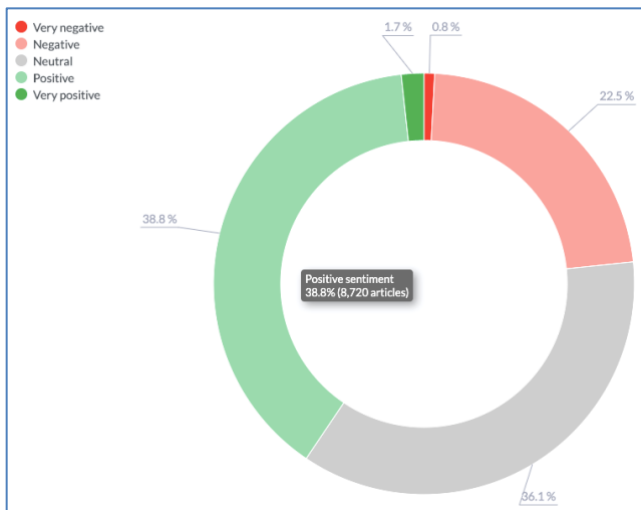
## 3 THE TREND OF CHILDHOOD OBESITY IN WORLDWIDE NEWS

In the following section we focus on a specific Public Health priority, the well-known epidemiological case of childhood obesity that is of a major EU concern. To do that we will use Event Registry (ER), customized by the MIDAS Horizon 2020 project, funded under ‘Big Data supporting public health policies’ to develop a Big Data platform that facilitates the utilization of healthcare data, making that data amenable to enrichment with open and social data [1]. The Event Registry system collects and annotates in real-time news articles published by over 100,000 news publishers worldwide [4]. It provides the user with public health news articles in more than 10 languages as well as world events mentioned in these articles, permitting to explore what is currently being reported about in the media worldwide. ER can (a) identify and download news content from publicly available news sources, (b) analyze and semantically enrich the articles regardless of the language, (c)

combine the articles that report about the same event into a single event, (d) extract the relevant event information and (e) make all information searchable [7]. The worldwide health monitoring potential of this tool was discussed in [12] in the context of public health decision-making support, in particular as an epidemic / pandemic intelligence tool [15].



**Figure 4: The concept trends associated with the query “childhood obesity” over the worldwide news in the past three years, highlighting related topics such as cancer, diabetes, nutrition, junk food or sugar.**



**Figure 5: The sentiment analysis of worldwide news relating to “diabetes mellitus type 2”, with a 38.8% of positive sentiment identified highlighting the good results of the continuous efforts done to overcome this problem of modern society.**

The user of ER can, query the dataset of worldwide news on “childhood obesity” to explore the trends and major concepts related with this query through the mentions in worldwide news (see Figure 4). Trending information is computed by comparing how frequently individual concepts/categories are mentioned in the articles. By default, trends are computed by comparing the total number of mentions of a concept in the last two days compared to the number of mentions in the two weeks before. The trend for each concept is computed as the Pearson residual. The returned concepts are the ones that have the highest residual [9]. The sentiment analysis in Figure 5 over the topic “diabetes mellitus type 2” serves

as a base of discussion to another angle on other aspect of news bias in the health domain: the sentiment expressed over news on a disease. The example shows that, although the immediate negative sentiment over such a topic, the positive sentiment can be identified in the good results from the continuous effort on fighting such a modern societal problem. In Figure 6 we show a screenshot of the pie chart of categories of worldwide news associated to the query “childhood obesity”. In that visualization module we observe that the media coverage of school meals nutrition in the context of public health was only 0.36% of all the news about childhood obesity during 2018. On the other hand, there was a coverage of 4.42% on child welfare in the context of the Society category, while the coverage on the business about sweeteners was 0.94%.

## 4 IMPACT IN PUBLIC HEALTH DECISION MAKING

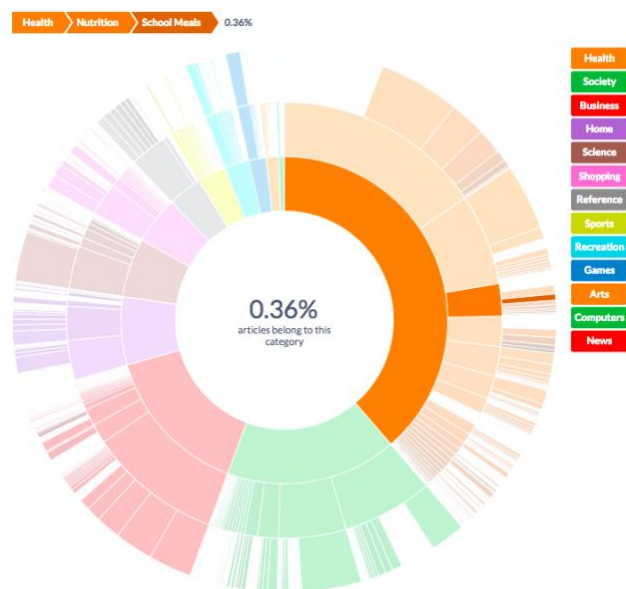
Research in the field of automatic detection of news media bias shifts its attention to sentiment analysis and opinion mining in the news. Most sentiment and opinion mining analysis has been done on very subjective texts like product launch reports, movie reviews or blogs, where the opinion of the author is expressed freely in a very subjective and biased way. Recently, sentiment analysis of news articles, where an opinion of a journalist should not be present, is getting more attention. An example of such an approach is the news media bias analysis of finding over and under-stated facts of a particular news outlet [11].

In the example of Figure 1, the event of the Ebola outbreak is identified immediately after the news articles that report about it are collected. One can explore the evolution of the news publishers awareness of the epidemics in a timeline by looking at the related news articles represented in a world map, as they were identified or updated during a selected period of time. ER can find articles and events related to a particular entity, topic, date, location or category, as well as measure their impact on social media (Twitter).

For each world event, ER is able to provide extensive information. Its event clustering permits us to distinguish between subtopics and perspectives in the stories relating to that particular event. Beside the whole list of articles that describe the event, the user can also see the list of top concepts, the trending of the articles, and subcategories it falls into. If a user is not interested in all events, (s)he can easily limit the list of articles and events displayed based on specific interests, location, etc.

In particular, Governmental Institutions are interested on what is the public opinion over public health related legislation such as the Sugar Tax, where sentiment analysis can be an approach with great potential. This and related measures were applied in several EU countries to fight against childhood obesity and consequential diseases such as diabetes mellitus type 2. The role of social media is of great importance in this context but also contributes with a lot of noise. A query on “Sugar Tax” in ER allows us to identify not only the news articles about this measure but also the social media mentions of those news items, permitting the user to estimate the impact of the issue in the population.

A new version of ER is in development, integrating an automated annotator assigning MeSH Heading descriptors to snippets of free text provided by the user. This will allow the annotation of the news articles with those useful classes, designed to enhance the exploration of biomedical research in the well-established search engine PubMed. The latter is part of the workflow and know-how of most health professionals. The new ER instance, built in the context of the MIDAS project will allow those health professionals to explore the worldwide and local news using the MeSH Heading descriptors much as they use them in their searches in PubMed. The new system will also provide MeSH Heading-based visualization modules such as the one discussed above and in Figure 6, providing an efficient perspective of the news coverage over subtopics of the search query, allowing for a fast identification of potential news bias, designed for the health domain, to support decision making in public health.



**Figure 6: The categories associated to the query “childhood obesity” over the worldwide news during 2018. This shows that the media coverage of school meals nutrition in the context of public health was only 0.36% of all the news about this topic.**

## 5. CONCLUSIONS AND FURTHER WORK

Media bias is a universal concern. Despite the fact that newspapers and reporters or journalists are supposed to provide the readers with impartial, objective, unbiased and reliable information, the reality is somehow different. Every news story has a potential to be biased. Every news story has a potential to be influenced by the attitudes, cultural background, political and economic views of the journalists and editors. In this paper we have discussed several angles of the news media bias in the context of health-related news with a particular focus on epidemiology. We have also presented some approaches and tools that permit data exploration and can help balancing the information in worldwide media coverage.

This includes some of the ER visualization modules which can help us to explore what is the news coverage of a certain health-related measure feeding the general population awareness. Moreover, the upcoming Event Registry instance built in the context of MIDAS will be offered with an automated MeSH classifier of text. This open source service will be used to classify the news with the MeSH headings and will enable queries using these. That will permit researchers to be closer to the information they are looking for, using a similar workflow as the one used in queries over PubMed. That will be useful to health professionals in particular that use PubMed daily in their biomedical research, and fully understand the usability of the MeSH descriptors. We will further analyze news bias when the first results of early adopters are available. Further work also includes the bias detection through advanced text mining techniques. This includes the analysis of the used metrics that can be itself a potential bias generator.

## ACKNOWLEDGMENTS

We thank the support of the European Commission on the H2020 MIDAS project (GA nr. 727721).

## REFERENCES

- [1] Brian Cleland et al. 2018. Insights into Antidepressant Prescribing Using Open Health Data. Big Data Research doi.org/10.1016/j.bdr.2018.02.002
- [2] D. Brossard et al (2013). Science, new media, and the public. Science 339.6115: 40-41.
- [3] J. Ginsberg et al (2009). "Detecting influenza epidemics using search engine query data." Nature 457.7232: 1012.
- [4] M. Grobelnik and G. Leban. Eventregistry's health panel. eventregistry.org. Accessed: 2019-08-22.
- [5] D. Kahneman et al (1982) Judgment Under Uncertainty: Heuristics and Biases. New York: Cambridge University Press.
- [6] J. Leask et al (2010). Media coverage of health issues and how to work more effectively with journalists: a qualitative study. BMC public health 10.1: 535.
- [7] G. Leban, B. Fortuna, J. Brank, M. Grobelnik (2014). Event Registry – Learning About World Events From News, WWW 2014, pp. 107-111.
- [8] C. H. Hennekens, J. E. Buring (1987). Epidemiology in Medicine, Lippincott Williams & Wilkins.
- [9] C. Manning et al. (2008). Introduction to Information Retrieval. Cambridge University Press 2008, pp. 269-273.
- [10] L. J. Martin, X. Biying and Y. Yasui (2014). Improving Google flu trends estimates for the United States through transformation. PLoS one 9.12: e109209.
- [11] S. Park, S. Kang (2009). "NewsCube: delivering multiple aspects of news to mitigate media bias." Proceedings of the 27th international conference on Human factors in computing systems.
- [12] J. Pita Costa et al. (2017). Text mining open datasets to support public health. WITS 2017 Conference Proceedings.
- [13] P. Snyder et al (2010). Science and the media: Delgado's brave bulls and the ethics of scientific disclosure. Academic Press.
- [14] WHO (2019). Ebola outbreak in the Democratic Republic of the Congo declared a Public Health Emergency of International Concern - News release 17 July 2019, <https://www.who.int/news-room/detail/17-07-2019-ebola-outbreak-in-the-democratic-republic-of-the-congo-declared-a-public-health-emergency-of-international-concern> Accessed: 2019-08-22.
- [15] WHO (2017). Epidemic intelligence definition. [www.who.int/csr/alertresponse/epidemicintelligence/en/](http://www.who.int/csr/alertresponse/epidemicintelligence/en/). Accessed: 2017-09-05



# Latent distance graphs from news data

Luka Bizjak  
Artificial Intelligence  
Laboratory  
Jozef Stefan Institute  
Ljubljana, Slovenia  
l.bizjak@student.fmf.uni-lj.si

Miha Torkar  
Jozef Stefan International  
Postgraduate School  
Artificial Intelligence  
Laboratory  
Jozef Stefan Institute  
Ljubljana, Slovenia  
miha.torkar@ijs.si

Aljaž Košmerlj  
Artificial Intelligence  
Laboratory  
Jozef Stefan Institute  
Ljubljana, Slovenia  
aljaz.kosmerlj@ijs.si

## ABSTRACT

Network analysis is one of the main topics in modern data analysis, since it enables us to reason about systems by studying their inner relations, for example we can study a network by analyzing its edges. However, in many cases it is impossible to detect or measure the network directly, due to noisy data for example. We present a method for dealing with such systems, more concretely we present a probabilistic model called latent distance network, which we use to model news data from **EventRegistry**. In the end of the article we also present experimental results on predictions of latent distance model with methods of machine learning.

## 1. INTRODUCTION

News articles offer a constant stream of new information about business activities, political events, natural disasters and a variety of other topics. Finding structure in such data is inherently difficult, because of the high levels of noise, repetitions and lack of structure. In order to systematically extract useful information from news, [6] developed a system called **EventRegistry**. It is able to collect news articles from various sources and languages, group them together according to their content and then extract relevant information from the texts about the grouped articles (entities, people, locations, topics). The groups of articles about the same content are called events and in this work we will explore the structure of connections between various topics of such events. Namely we will build a network of connections between topics that appear in the news events, track the evolution of connections through time and predict future relationships.

This will be done through the framework of latent distance graphs, because each event will be placed in an embedded space according to its content. The distance to other events

in this space will then represent the similarity between them. Latent distance graphs are an example of metric random graphs, where the probability of connection between two nodes is dependant on their position in the space. These types of graphs are also called network models, which represent multi-dimensional data. In our case the transformation of the events from EventRegistry will yield vectors in a 300 dimensional Euclidean space that are then used for calculation of a distance function that determines probability of connection between nodes. The procedure to extract a network of connections from the events data can be generalized, because one can apply it in any case where the distance between objects is well defined.

The rest of the paper is organised as follows, section 2 describes the news data that we were using. In section 3 we introduce the word embeddings that are used to obtain vector forms of news events. In section 4 we explain the latent distance model. section 5 presents the analysis done on the graphs and section 6 concludes by pointing out the main results, stressing some difficulties of our approach and giving directions for further work.

## 2. DATA

We have worked with the news data from the EventRegistry and in particular, we download all of the news events from business category in years 2017 and 2018. This yields the events overall and it was obtained with python package **EventRegistry**. Every event that was obtained contains information about which entities were involved and **EventRegistry** also provides classification of events into certain categories as defined in the *dmoz* taxonomy. These categories are structured hierarchically and are divided into various sub-categories, where, for example, category "Business" is split further into "Banking and services" and then into "Investing". For further details see [6].

## 3. WORD EMBEDDINGS

Here we give a brief overview of how the word (word2vec) embeddings are being calculated, for detailed explanation we advise reader to consult [9]. One of the algorithms which can calculate the word embeddings is called word2vec and it can be trained by one of the sub-algorithms, called Continuous-Bag-Of-Words (CBOW) and Skip-Gram (SG). We give more precise explanation of the Skip-Gram algorithm since we use

it in the process of generating the latent model.

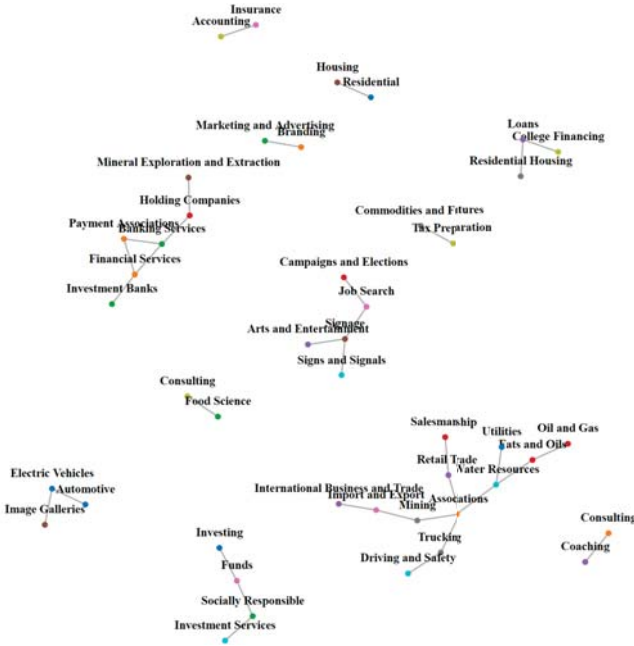


Figure 1: Latent distance graph of the Business category

### 3.1 Skip-Gram

The Skip-Gram (SG) sub-algorithm trains a shallow neural network (NN) to learn the vector representation of each word in the sentence from its surrounding context words. Namely for each word  $w_n$ , the neural network predicts the surrounding context words  $Con$ , where the user has to define the number of surrounding words that are being predicted. For example if  $Con = 2$ , we predict the pair  $(w_{n-1}, w_{n+1})$ . The NN that is being trained has only one hidden layer between the input and output layer, which in practice gives two transformations of the input vector. The first one from the input to the hidden layer and the second one from the hidden layer to the output. Hence we have the target word at the input layer of the NN and the context words are at the output layer.

In order to give more detailed overview of the algorithm we need to set some notation first. Let  $x$  be a word in vocabulary and let  $N$  be the size of the vocabulary, so  $x_1, \dots, x_N$  are all the words in the vocabulary. Then let  $K$  be the dimension of the hidden layer and  $C = Con$  be the number of context words. Also denote by  $W$  the  $N \times K$  weight (transformation) matrix. The equation for hidden layer is in this case then

$$\mathbf{h} = W^T x = W_{k,.}^T, \quad (1)$$

where  $W_{k,.}$  is the  $k^{th}$  row of the transformation matrix  $W$ . From the hidden layer to the output layer we apply another transformation matrix that is denoted by  $W'$ . The output layer in this case then consists of  $C$  multinomial distributions and we use the matrix  $W'$  to calculate the score vector  $\mathbf{u}$  for the  $j^{th}$  unit on  $c^{th}$  context word as follows:

$$u_{c,j} = u_j = (W'_{.,j})^T \mathbf{h} = (W'_{.,j})^T W_{.,k}^T, \quad (2)$$

for all  $c = 1, 2, \dots, C$ . The final output is the given by:

$$\mathbb{P}(w_{c,j} = w_{out,c} | w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{i=1}^N \exp(u_i)}, \quad (3)$$

here  $w_{c,j}$  is the  $j^{th}$  word on the  $c^{th}$  panel of the output layer and  $w_{out,c}$  is the true  $c^{th}$  word among the output context words, thus  $y_{c,j}$  is the final output of the  $j^{th}$  unit on the  $c^{th}$  panel. The authors of both CBOW and SG [8] have noted that SG is slower than CBOW but produces better results for infrequent words. For more detailed exposition please see [9].

### 3.2 Events to vectors

In order to obtain vector forms of events, we use all of concepts and their weights that are extracted from the news article of every event by EventRegistry. In particular the vector form of each event is calculated as

$$\text{Event}_i \equiv e_i = \sum_{c \in C_i} w(c) \cdot \text{word2vec}(c), \quad \forall e_i \in E, \quad (4)$$

where  $C_i$  represents the set of all concepts of the event  $i$  and  $w(\cdot)$  gives a value of the weight of the concept in the event in the interval  $(0, 100]$ . Each event is then represented as a numeric vector in  $\mathbb{R}^N$ , where  $N$  is dependant on the dimension of the word2vec space. In our case we choose  $N = 300$ , since we are using the pre-trained word2vec model by Google.

## 4. LATENT DISTANCE NETWORK MODEL

Latent distance networks or graphs can be considered as particular examples of random graphs ([2]). Random graph consists of the set  $V$  of vertices or nodes, set  $E$  of edges. It is normally denoted by  $G(n, p)$ , where  $n = \text{card}(V)$  and  $p$  is the probability of an edges between two vertices. One of the most basic examples of random graphs is the Erdos-Renyi graph  $G_{ER}(n, p)$ , which is defined by saying that each edge is included in the graph with probability  $p$  independent from every other edge. Having this example in mind, we can define random graphs with  $n$  vertices by giving probability distribution for edges, this is construction carries over to latent distance graphs, where probabilities for edges are derived from probability distribution of distances between vertices.

The latent distance graph is represented by  $N \times N$  adjacency matrix. The latent distance model is given as follows, we first define a distance function on  $\mathbb{R}^N$  by:

$$d(x_k, x_{k'}) = \rho e^{-\frac{\|x_k - x_{k'}\|^2}{\tau}}, \quad (5)$$

where  $\rho$  represents the sparsity of the network and  $\tau$  represents characteristic distance scale. Each vertex of the network is at this stage represented by some vector  $x_k \in \mathbb{R}^N$  (as describe in 3.2). To get a random model we need to specify some probability distribution of distances between vertices, which will give us the probabilities for edges between vertices. These then correspond to the adjacency matrix of our network and in the case of the latent distance graph it is given by:

$$A_{k,k'} \sim \text{Bern}(d(x_k, x_{k'})). \quad (6)$$

Thus the entries of the adjacency matrix  $A$  are given by Bernoulli distribution of distances between vertices of the

graph. Such networks were also considered in [7] in connection with Hawkes processes defined on networks. Note that one could choose any other appropriately normalized metric i.e. taking values in  $[0, 1]$ , on  $\mathbb{R}^N$  and the construction would work as well. Moreover in Erdos-Renyi graph  $G(n, p)$  the sharp threshold for the connectedness is given by  $\frac{\ln(n)}{n}$  (see [4]). To the best of our knowledge no such sharp threshold is known for the latent distance graphs.

## 4.1 Generating the latent model

In this section we describe how we generated the latent distance graph from news data.

### 4.1.1 News data latent model

After we obtain the numeric forms of the events via the procedures described in section 3 we are able to form a latent model on top of news data. The embedded events now correspond to vectors in  $\mathbb{R}^N$ , which gives us a finite set of vectors  $(x_i)_{i=1}^K \subset \mathbb{R}^N$ . We can then construct the weight matrix, which is basically the matrix of distances between different pairs of points,

$$W = (w)_{i,j}^K = (d(x_i, x_j))_{i,j}^K, \quad (7)$$

where  $d(., .)$  is the distance function (5). Once we have the weight matrix we can generate the latent distance graph by defining matrix of probabilities as the adjacency matrix:

$$p_{i,j} = \text{Bern}(W_{i,j}), \quad (8)$$

where  $p_{i,j}$  represents the probability between two nodes  $x_i$  and  $x_j$ .

## 5. GRAPH ANALYSIS

We perform some basic analysis on latent distance graphs derived from news data. We generated the adjacency matrices for each day in a one year time period, thus we get 365 graphs. Each represents the activity of the events from the business category. We also perform clustering of the events into 100 most frequent subcategories interacting with business category. We perform this so that we replace the adjacency matrix which depends on number of events we consider, say  $K$ , with a matrix depending on fixed number of parameters and make an aggregated distance matrix  $W_{agg}$ , which we define as follows:

$$(W_{agg})_{k,l} = \sum_{i,j}^K \mathbf{1}_{c_k=i, c_l=j} w_{i,j}. \quad (9)$$

Now we can generate a new graph whose nodes now represent categories under consideration. Example of such a graph is given in Figure 1. Then we can generate adjacency matrices of these graphs for a fixed time period, one year in our case.

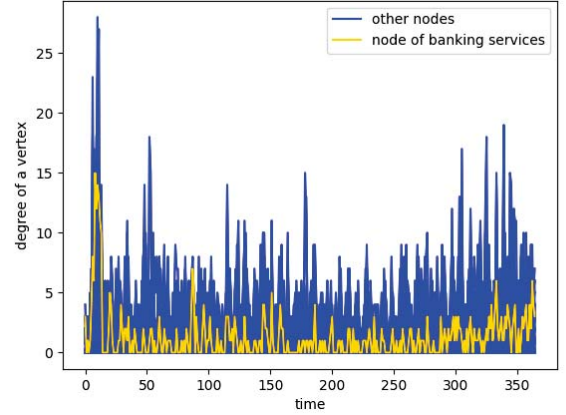
### 5.1 Graph evolution

Now that we have sequence of graphs  $\{G_1, G_2, \dots, G_{365}\}$ , we can view this sequence as evolution of the network in given time frame. Thus we can check the interaction of category  $i$  in  $\{G_1, G_2, \dots, G_{365}\}$  by just summing over all the adjacency matrices and looking into appropriate row. For example in the Table 1 below we show interaction of subcategories **Banking and services** and **Oil and Gas**. To be concise we only show top five interacting subcategories.

We also computed the degree of the nodes in  $\{G_1, G_2, \dots, G_{365}\}$  and plotted the time series of each node, see Figure 2. From Figure 2 we can see that there is some change in the degrees through time and this opens up a possible direction to study such networks dynamically.

Dependencies between categories					
Fixed category	Cat1	Cat2	Cat3	Cat4	Cat5
Banking and services	Holding Companies	Financial Services	Finance	Payment Associations	Investment Banks
Oil and Gas	Fats and Oils	Mining and Drilling	Import and Export	Payment Associations	Job Sharing

**Table 1: Dependencies of categories in the dynamical network**



**Figure 2: Degrees of nodes through time**

### 5.2 Predictions

We use neural network model to make predictions about potential structure of the network in the near future. Specifically we build the model on the top level categories i.e. **Business**, **Politics** and others. We use the aggregation process (9) to generate all the adjacency matrices for all levels, which can then be put into vector form, so that we can then use them as inputs for LSTM Neural Network model [5]. For the model we used LSTM Neural Network, where we used three residual LSTM layers and final dense layer. We optimized with mean squared error (MSE) with ADAM optimizer [5]. The results of the experiments are displayed in Figure 3 and Figure 4.

## 6. CONCLUSIONS AND FUTURE WORK

In this work we collect data from EventRegistry about all business events from years 2017 and 2018 and build a latent distance model on top of it. We are able to do this by transforming the textual news data to numeric vector forms through word embedding algorithm word2vec. The latent graph model that is produced in this manner gives us a reasonably good representation of EventRegistry data as

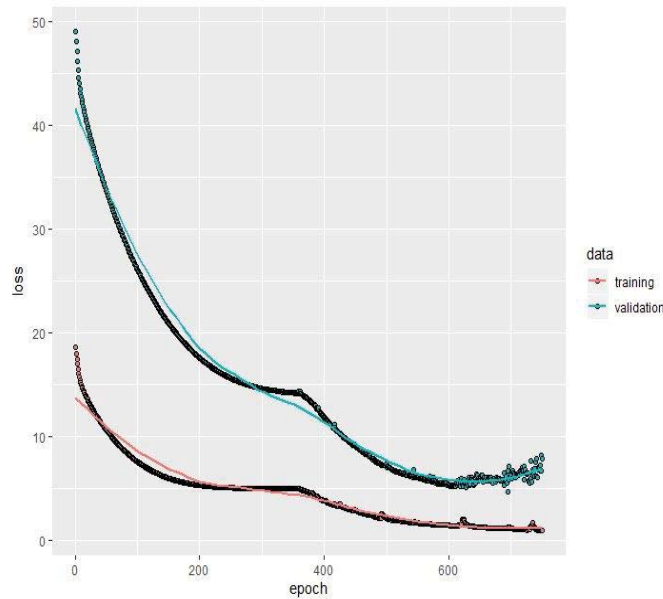


Figure 3: MSE of learning on whole Neural Network

can be seen in Figure 1. However we believe that this model could be used on other similar data sets as long as difference between object can be defined with some metric. In order to avoid issues with noise we used a more compact representation of events, where we clustered them into a predefined set of categories. In this compact form the adjacency matrices of graphs were then fed into a LSTM model for prediction of how the graph will evolve in the next step. The results of this part were reasonably good and can be seen in the Figure 3 and Figure 4.

Let us now point out some difficulties of this approach. The first one is that it seems that the representation depends to some extent on word embeddings that are used. This can be seen from example in Table 1, where we have strong connection between **Oil and Gas** category and subcategory **Fats and Oils** which should not be connected. The second problem is the sparsity of the adjacency matrices, which makes it very hard to achieve good performance with machine learning techniques as well perform spectral analysis [1] on the adjacency matrices. This last point is connected to theory of dynamical graphs, where our presented sequence  $\{G_1, G_2, \dots, G_{365}\}$  serves as one example. In future work we would like to try to extend the LSTM model from above to predict lower level categories as well. This would be done in several steps where each level would be predicted after the previous one. Finally we would like to understand how to resolve the sparsity problem (some techniques for dealing with this problem are presented in [3]) and then apply techniques from dynamical graphs [1], in particular we would like to know spectral distortions [1] of these graphs.

## 7. ACKNOWLEDGMENTS

The research project leading to these results received funding from the European Union Research and Innovation programme Horizon 2020 under grant agreement No. 675044

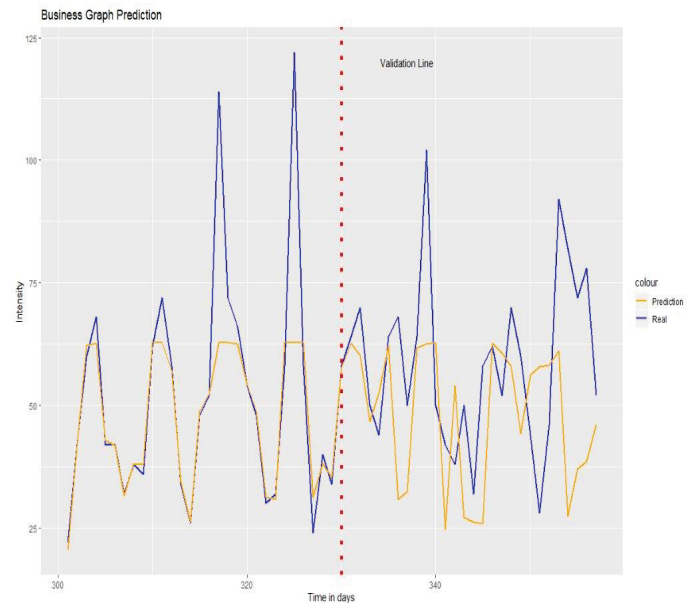


Figure 4: Training and prediction scores for Business category

(BigDataFinance). We also wish to thank Jakob Jelenčič for his help on implementation of prediction models and M. Beshar Massri for the help with the graph visualization.

## 8. REFERENCES

- [1] L. C. Aleardi, S. Salihoglu, G. Singh, and M. Ovsjanikov. Spectral measures of distortion for change detection in dynamic graphs. In *International Conference on Complex Networks and their Applications*, pages 54–66. Springer, 2018.
- [2] B. Bollobás and B. Béla. *Random graphs*. Number 73. Cambridge university press, 2001.
- [3] E. J. Candès. Mathematics of sparsity (and a few other things). In *Proceedings of the International Congress of Mathematicians, Seoul, South Korea*, volume 123. Citeseer, 2014.
- [4] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.
- [5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- [6] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 107–110. ACM, 2014.
- [7] S. Linderman and R. Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning*, pages 1413–1421, 2014.
- [8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [9] X. Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.



# Document Embedding Models on Environmental Legal Documents

Samo Kralj  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
samo.kralj1@ijs.si

Erik Novak  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
erik.novak@ijs.si

Živa Urbančič  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
ziva.urbancic@ijs.si

Klemen Kenda  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
klemen.kenda@ijs.si

## ABSTRACT

Finding similar documents in a big document corpus based on context has many practical applications especially in the legal sector. In this paper, our focus is on the documents related to environmental law which have been collected in a database of approximately 300k documents. We analyzed the performance of different representation models (called document embeddings) on our database and found that evaluating the results is difficult, due to the size of the database. The approaches presented can be applicable for other text datasets.

## Keywords

text analysis, natural language processing, environmental law, machine learning, word embedding, document embedding

## 1. INTRODUCTION

When working with a large number of documents one can perform different tasks, such as finding patterns and topics within a documents, labeling documents based on their content, and finding documents that are similar to each other. These tasks can be found in multiple domains - one of them being the legal domain. There, lawyers spend hours finding documents and parts of these documents to support their legal cases.

In this paper, we present our preliminary results for finding similar documents. We employ word embeddings for creating different document representations - called document embeddings. The goal is to construct a document embedding model that enables the user to quickly find documents that are similar to a user chosen document. The documents used for evaluation are from the legal domain, but the approach can be applied to more general text datasets.

The remainder of the paper is as follows. Section 2 describes the data sources used for creating the document embeddings. Next, section 3 presents the content extraction and enrichment tool used for extracting additional docu-

ment metadata. In addition, it describes different models of document embeddings using the pre-trained word2vec and fasttext word embedding models, as well as our word embedding model trained exclusively on the collected environmental documents. Section 4 presents the preliminary results of the document embedding analysis, followed by the description of future work in section 5. We conclude the paper in section 6.

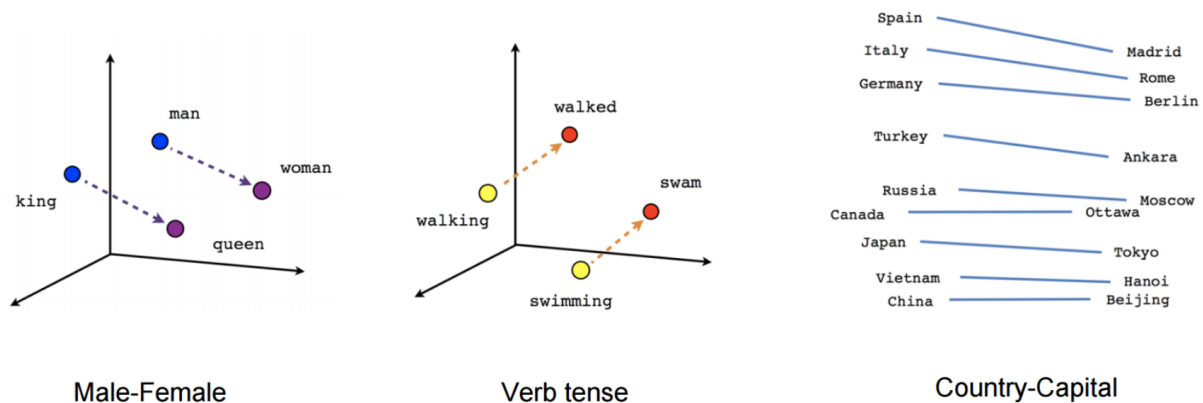
## 2. DATA

The legal datasets used for the analysis were collected from two main sources: the first is ECOLEX [1], an online information service on environmental law led by Food and Agriculture Organization of the United Nations (FAO), the International Union for Conservation of Nature (IUCN) and United Nations Environment Programme (UNEP). The second dataset was acquired from EURLEX [3], a database of entire European Union law.

### 2.1 Data Acquisition

The data was collected using dedicated web crawlers. In particular, we attempted to collect as much information about each document as possible. In total, 220k and 800k different legal documents are available on ECOLEX and EURLEX datasets, respectively. The documents are ranging from the start of 20th century up until the year 2019.

There is much document metadata which is available for documents from both sources, such as the document's title, its authors, various dates (i.e. day of proposal, the day it went into force, etc.), the subject of the documents and various keywords (which are called "descriptors" in the EURLEX dataset). Bearing this in mind, there are many differences between the two acquired datasets. In this article we focus on the following: the ECOLEX dataset consists entirely of environmental law. In addition, the dataset contains much more metadata, including geospatial information (i.e. locations and countries affected by the given document), as well as a short abstract. On the other hand, the EURLEX dataset contains less metadata, but provides the complete



**Figure 1: Word relationships captured by word embeddings. They are able to identify different relations such as male - female terms, verb tenses and other.**

document content in raw text for most cases in the dataset.

Additionally, the datasets are different in two important metadata attributes: the keywords in the ECOLEX dataset and the descriptors in the EURLEX dataset. These keywords are words or phrases that best describe what the document is about. It is to be expected that documents that have similar keywords are also similar in content. While keywords and descriptors serve similar purpose in the respective dataset, they are not the same. A particular keyword might not be included in the descriptors word corpus and vice versa. Furthermore, keywords that describe some document are different from descriptors that describe a similar document.

## 2.2 Dataset Statistics

Out of the 800k EURLEX documents collected, 300k were filtered out based on whether the full text of the document is available in English, German and Slovene language. Since we are interested in documents dealing with environment, further filtering was done using document descriptors, keeping only documents with at least one environmental descriptor. In total, approximately 75k documents were considered to be appropriate for our analysis.

Since the ECOLEX documents are already focused only on environment, no further filtering was required.

## 3. METHODOLOGY

In this section we describe our approach for analyzing a big corpus of documents, namely document embeddings. Even though a lot of pre-processing was necessary to prepare the documents' texts for later use (making sure all letters are lowercase, stripping the punctuation from the text, removing words that appear frequently in the language – for example prepositions), we will not discuss this further in the paper. We also appended additional information to the documents using the content extraction and enrichment tool, which we describe in section 3.1. Further, we focus on word embeddings and different methods of how to use them to create document embeddings in sections 3.2 and 3.3, respectively.

### 3.1 Content Extraction and Enrichment Tool

To enrich the documents, we annotated all documents using the InforMEA ontology, a hierarchy of environmental terms. Also, document's text was sent into Wikifier - a web service that extracts major Wikipedia concepts from the text. The resulting concepts were added to the document's metadata. These annotations add additional keywords and concepts to the document, improving our representation of documents that may have poor keywords representation, and adds additional metadata to documents that already had a good keyword representation but might be missing some important keyword.

### 3.2 Word Embedding

In natural language processing, word embedding has been a popular method for representing textual data in the past years. It is a model trained on character n-grams of the word and on what is called context: the target word's neighboring words. In the model, the words are represented as vectors – usually in high-dimensional space - where the inherited geometric relations mimic relationships between words in the language. Word embeddings are able to capture both syntactic and semantic information about the word. Some of the relationships between words captured by word embeddings are shown in figure 1.

The most popular word embedding models available to the public are word2vec [10] and fasttext [9]. What sets them apart is what they consider to be an atomic embedding element: word2vec considers a word to be the smallest part of language to embed, while fasttext uses character n-grams as well - it embeds them as if they were words. Because of this we can extract embeddings for out-of-vocabulary terms, providing embeddings of rare and previously unseen words. We decided to employ two models: a) the pre-trained fasttext model for the English language, and b) the model trained on our database of environmental legal documents. In addition, aligned vectors for 44 languages [6, 4] are available, which will be used in the future work to enable cross-lingual search of documents.

### 3.2.1 Training a Word Embedding Model

One of the word embedding models we employed was trained on our database. Instead of having a large vocabulary of pre-computed word embeddings trained on Wikipedia and Common Crawl, this newly trained model is trained on documents from a more specific domain - resulting in a vocabulary limited to the topics found in the documents within the corpus (e.g. in our case environmental law). This approach might improve the performance in cases when the language is domain specific.

The new fasttext model has been trained using the gensim library. In order to be consistent with the pre-trained fasttext model, we decided the trained model should provide word embeddings as 300-dimensional vectors. We set a threshold of 4 appearances to avoid noise. In comparison with the vocabulary of the pre-trained fasttext model, the vocabulary of our model is 5 times smaller, consisting of approximately 500k tokens. Its initial performance is described in section 4.1.

### 3.3 Document Embedding

To be able to retrieve and compare documents, they must first be represented in a form that the machine will be able to understand. Similar as for words, the most common form of document representation is as a vector. We chose to represent a single document as an average of word embeddings of words found in that document. In other words, let  $W = \{w_1, w_2, \dots, w_n\}$  be a list of words that appear in a document, and let  $\{x_1, x_2, \dots, x_n\}$  be the list of word embeddings associated with the words in the document. Then the document embedding is calculated by following the equation

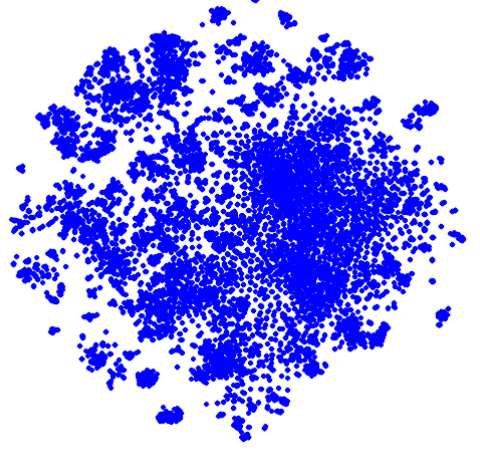
$$d = \frac{1}{|W|} \sum_{w_i \in W} x_i.$$

Further, we considered some other embedding methods. The first approach is to define the document embedding as an average of word embeddings of only the most significant words, namely document descriptors for the EURLEX dataset or keywords for the ECOLEX dataset. The reasoning behind it is that it might speed up the calculation, but it comes at a cost of neglecting a lot of information we have about documents and the possibility of reducing the quality of the result. To avoid the listed downsides, we propose a combined embedding, which would be defined as a linear combination of two embedding methods described above. This embedding unfortunately loses the advantage of fast computation, but it does give more weight to more important words of the document. In order to decide which method performs better, we performed some analysis which is described in section 4.

Once the document embeddings are calculated - depending on the chosen method and word embedding model - we are able to find semantically similar documents by calculating the distance of their embeddings. Figure 2 shows the mapping of the document embedding into the 2-dimensional space using the t-SNE algorithm [8].

## 4. PRELIMINARY RESULTS

We split our analysis in two parts. In section 4.1 we tested various document embedding models based on the choice of



**Figure 2: Planar projection of document embeddings of the first 15k English legal documents in the EURLEX corpus. Our assumption is that similar documents have similar embeddings and therefore form clusters, which we evaluated manually.**

word embedding models. In addition, we perform an analysis using different approaches of constructing document embeddings given a pre-trained fasttext word embedding model, which is described in 4.2.

### 4.1 Performance of Different Word Embedding Models

When deciding which document embedding model to use, the choice of word embedding model is very important. We are interested in which of the two word embedding models described in section 3.2 produces a better document embedding model. In this part of the analysis we chose to construct document embeddings as the average of word embeddings of words appearing in the text of the document.

Manually checking the results for some arbitrary examples we noticed that the newly trained word embedding model outperforms the pre-trained one when the source document is not particularly similar to any other document in the database. Our observations are based on using only the model trained with parameters described in section 3.2.1. Further analysis of training parameters will be performed in the future.

### 4.2 Performance of Different Document Embedding Models

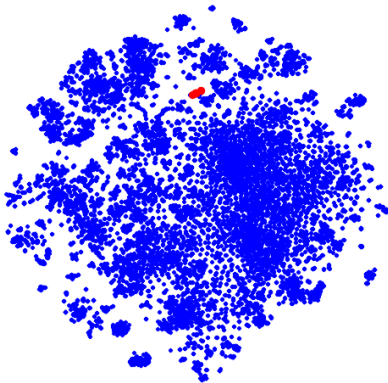
It is hard to evaluate and compare different document embedding models. We performed manual checking and found satisfactory results in some cases. To test the model we picked a random document and found the  $k$  "most similar" documents using the  $k$ -nearest neighbors algorithm [5] and the cosine distance.

What follows is an example of such a search for  $k = 5$  using a document embedding model based on the text of the document (the title is not included). The first item is the title

of the source document, while the rest are the titles of the most similar documents:

1. **Source:** European Convention for the protection of animals kept for farming purposes.
2. Convention on the protection of the Mediterranean Sea against pollution (Barcelona Convention).
3. Protocol concerning Mediterranean specially protected areas.
4. Protocol for the protection of the Mediterranean Sea against pollution from land-based sources.
5. Protocol of amendment to the European Convention for the protection of animals kept for Farming purposes.

The given results are quite good, but it seems like the document on the fifth position is the most similar to our source document - showing that the presented model still has potential for improvement. Figure 3 shows the result of the search for 10 most similar documents using the text of the document in document embeddings. Marked with the red dots are the documents acquired from the search results.



**Figure 3:** Projection of a document embedding model using the words from documents text as a representation. Red dots represent the 10 document embeddings that are closest to the embedding of the source document.

## 5. FUTURE WORK

Manually checking the complete corpus of a few 100k documents is time consuming. The amount of documents is huge and we also do not have the ability to tell how good the results are. There is no easy way to define a metric that could compare how well different models perform. Therefore, we will try to evaluate and improve our model using the users feedback. We will develop a service which will enable the user to perform queries for the legal documents. Each time a user makes a query, the system will note the documents that the user checked. With this feedback we will be able to update and improve our model.

In addition, we will consider another distance metric called the Word Movers Distance [7] when calculating the document similarity using word embeddings.

## 6. CONCLUSION

Word embeddings and document embeddings have proven to be useful when performing analysis on a large textual dataset. The available word embedding models on which we based our research - word2vec and fasttext - are exhaustive and easy to use. What we have done so far has given satisfactory results on recognizing similar documents, which we hope to improve with further work, especially by finding a model that will fit our dataset of environmental legal documents best and then developing it based on user feedback.

## 7. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the European Unions Horizon 2020 project enviroLENS under grant agreement No 821918 [2].

## 8. REFERENCES

- [1] Ecollex - a gateway to environmental law. [https://www.ecollex.org/result/?q=&xdate\\_min=&xdate\\_max=](https://www.ecollex.org/result/?q=&xdate_min=&xdate_max=). Accessed: 2018-12-20.
- [2] EnviroLens project. Accessed in: August 2019.
- [3] Eur-lex - access to european law. <https://eur-lex.europa.eu/homepage.html>. Accessed: 2019-02-25.
- [4] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
- [5] COVER, T., AND HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1 (January 1967), 21–27.
- [6] JOULIN, A., BOJANOWSKI, P., MIKOLOV, T., JÉGOU, H., AND GRAVE, E. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018).
- [7] KUSNER, M. J., SUN, Y., KOLKIN, N. I., AND WEINBERGER, K. Q. From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37* (2015), ICML’15, JMLR.org, pp. 957–966.
- [8] MAATEN, L. V. D., AND HINTON, G. Visualizing data using t-sne. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
- [9] MIKOLOV, T., GRAVE, E., BOJANOWSKI, P., PUHRSCH, C., AND JOULIN, A. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).
- [10] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (USA, 2013), NIPS’13, Curran Associates Inc., pp. 3111–3119.

# Local-to-global analysis of influenza-like-illness data

J. Pita Costa, F. Fuart,  
L. Stopar  
Quintelligence,  
Jozef Stefan Institute, Slovenia

D. Paolleti  
ISI Foundation,  
Italy

M. Hirsch  
German Research  
Center for AI (DFKI),  
Germany

R. Mexia  
Instituto Nacional de Saúde  
Doutor Ricardo Jorge  
(INSA), Portugal

P. Carlin  
South Eastern Health and  
Social Care Trust, UK

J. Wallace  
Ulster University,  
UK

## ABSTRACT

The need for appropriate, robust and efficient epidemic intelligence tools is increasing in this age of a connected society. Global health initiatives, such as Influenzanet, potentially have a central role in the future of Public Health. This paper presents the contributions to the Influenzanet initiative, describing a new monitoring system for local hubs and their data sources, based on Elasticsearch.

It is often the case that the exploration of internally generated data is prioritised by national public health institutions, and therefore cannot be addressed in the global Influenzanet platform. This platform can be used by health professionals without programming expertise to encourage and enhance their independence from busy in-house IT departments and further contribute to the effectiveness of their own research.

The most meaningful data visualization modules can then be considered for integration into the full Influenzanet platform that will serve the complete network, thus collaborating at a global level. With this approach we also show the importance that an active hub in carrying out its own investigations towards its own priorities. In that regard and as an example, we also describe new results on the application of state-of-the-art approaches to a local data set, using the Portuguese ILI seasons between 2005 and 2013. This study is based on the application of the Streamstory approach. It aims to show the potential of this versatile approach in: (i) identifying data-driven ILI seasons; (ii) relating the ILI incidence to the dimensions of weather data; and (iii) comparing the incidence throughout four different ILI definitions.

## CCS CONCEPTS

• Real-time systems • Data management systems • Life and medical science

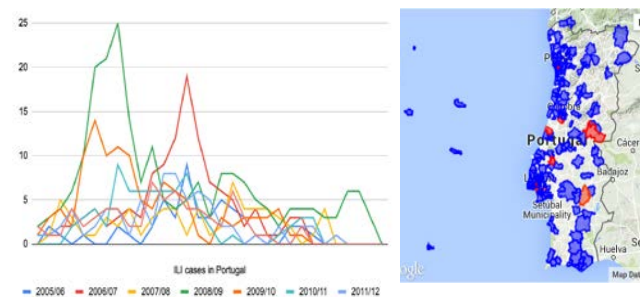
## KEYWORDS

Public health, Influenzanet, ILI, Elasticsearch, Streamstory

## 1 Introduction

With the recent worldwide threats to health being reported throughout the media, the need for efficient epidemic intelligence tools is paramount. It is important to note that the influenza virus is also part of these epidemic threats requiring monitorization, despite its less mediatic weight. The speed of mutation of the virus makes any epidemic unpredictable. Its socioeconomic impact is evident in

the number of workplaces affected every year during the season and the associated mortality in particular demographic groups (very young, very old). Influenzanet is a participatory surveillance monitoring system based on volunteers, submitting an online symptom questionnaire on a weekly basis, this enables a real-time global view of the incidence of influenza-like illness (ILI) across Europe. Note that the confirmation of influenza virus would require biological evidence and, thus, (often expensive) sample collection. The data set is collected in real time by the Influenzanet system and each volunteer provides a profile survey (including important information such as being a smoker, usual transport, etc.) and the weekly questionnaire of symptoms (see an example of the latter in Figure 1). The latter gathers the information that permits the identification of the presence of ILI that can be defined in at least four different ways according to the symptoms considered [6].



**Figure 1: Incidence of influenza in Italy, between 2009 and 2013, collected by the local Influenzanet hub Gripenet.pt**

## 2 Monitoring and exploring the local data

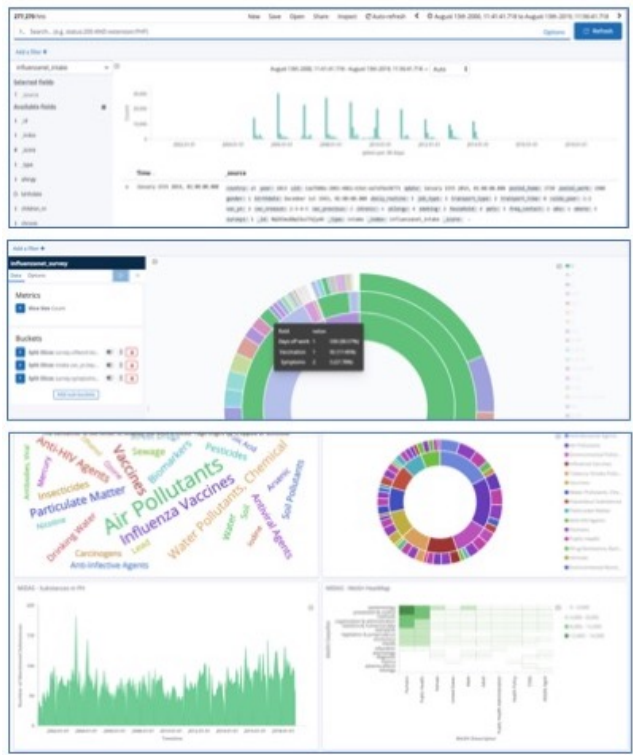
The Influenzanet system is deployed in more than ten European countries working in parallel under the same IT data collection framework, with some variety in the focus of the ILI monitoring [7]. This is usually aligned with the local public health priorities and ongoing studies, where most of the work is done by health experts, some of whom have some data science know-how or else are backed-up by in-house technical support.

To guarantee some independence to the less technical staff we have developed a data visualization dashboard that provides the user with real-time access to a local data set sourced on the national volunteer participants. This is based on Elastic Search technology, together with the Kibana open source data visualization plugin. Part of this work was developed in the context of the European Union research project MIDAS [3], by applying the know-how obtained



in building a similar system to monitor and manage the scientific knowledge open data set MEDLINE [5]. Note that the latter can be used to provide complementary information and be deployed in parallel to the Influenzanet dashboard.

The local Influenzanet data can be delivered to the dashboard through an API to the main platform. The update of the back-end system is driven by import scripts that appropriately load the new dataset into a new index in Elastic Search. This new Influenzanet-local index (comprised of one for surveys and the other for the symptom questionnaires) generates the database that serves the monitoring system. The dedicated dashboard based on Kibana has a native integration with Elastic Search and, therefore gets the index imported automatically to dynamically build the new visualization modules and dashboards. The public instance that can be derived from a dashboard is dependent of the choices in the definition of that dashboard.



**Figure 2: Dashboard of visual modules to monitor KPIs at each local Influenzanet hub, based on Elasticsearch and Kibana**

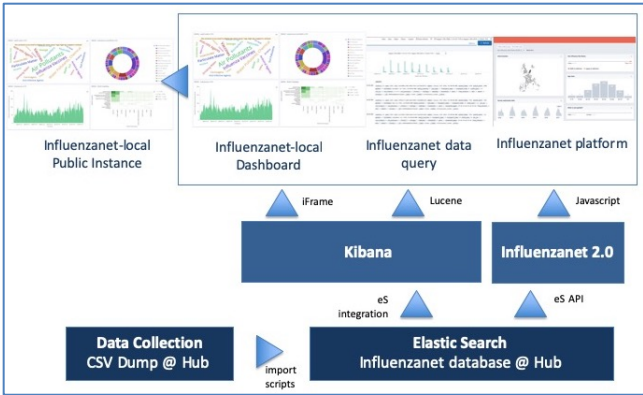
It is often the case that specific interests in the local exploration of the Influenzanet data arise, directly related to the local public health priorities and ongoing studies. The possibility to explore their own local data through a technological tool offered as a service, can be of great value to the Influenzanet hubs (i.e., national Influenzanet-related institutions that collect the data). Although the data must be collected using a homogeneous approach to enable overall comparisons, the exploration of that data can target specific aims. The Elasticsearch-based system presented in this paper empowers the user with less technical expertise to build data visualization

modules from subsets of the dataset that correspond to their own KPIs (Key Performance Indicators) they wish to monitor. This service will support an evidence-based policy-making by the national public health authority. The following are example queries made over the example data visualisation modules available in the Influenzanet dashboard:

- What are the most prominent symptoms per year?
- What is the coverage of Influenzanet surveys? (counts of questionnaires per country/year)
- When in the year the symptoms are more prevalent?
- What is the relationship between the incidence of ILI, the days at work and taking the Influenza vaccine?

The technical independence from the often busy IT departments enables health professionals to go further and faster in the exploration of their data through interactive visualization modules displayed through a dynamic dashboard (see Figure 2).

The architecture within the system relies on two useful tools provided by the Kibana technology (see Figure 3). The data collection is loaded by the local Influenzanet hub and is immediately made available at the Influenzanet data query dashboard, where the parameters of the data can be easily accessed and manipulated to subset the data or to produce powerful Lucene-based queries. With the saved subsets of data the user can create interactive visualization modules that will then integrate with the monitoring dashboard.



**Figure 3: The architecture of the Elasticsearch-based system that enables the visualization of local own data at each Influenzanet hub**

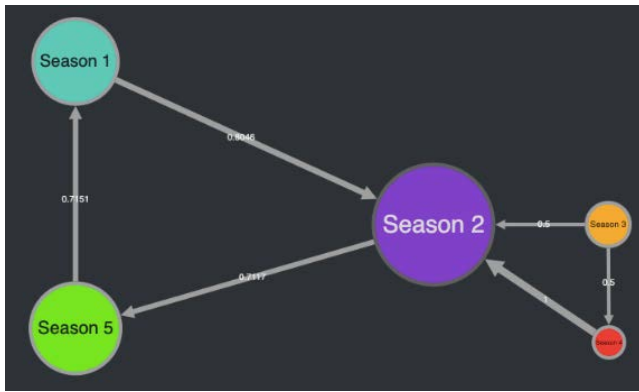
The Influenzanet network is preparing version 2.0 of its platform [6] that includes a modern architecture, making use of public APIs and storing the data locally, separating it by user data, survey data, and content data [4] [1]. The new Influenzanet platform will provide the Influenzanet hubs with a common set of data visualization modules. The plan is to augment “classical” Influenzanet data collection with additional sensor data from mobile phones [2]. Moreover, it includes a micro service architecture based system for better scalability and a more flexible development process. The backend of the new platform will be offered as Software as a Service (SaaS) but can also be downloaded as a self-hosted version. It will leverage this service in the

perspective of having access to the most meaningful data visualization modules throughout the Influenzanet hubs. The latter will be evaluated and can be integrated if they represent common value to other members of the Influenzanet network.

There are an ever increasing number of data sources that potentially could be used to gain new insights into areas such as disease prevention and policy formulation/evaluation, but these are not optimised for use within a data analytics type user interface. The MIDAS project was funded under a call for ‘*Big Data supporting Public Health policies*’ to develop a big data platform that facilitates the utilisation of healthcare data beyond existing isolated systems, making that data amenable to enrichment with open and social data. This aligns closely with the efforts in Influenzanet, and the research work we have developed uses 5 year sample of this dataset. For this reason we made available a live demo page with videos and demos that can be shared with Influenzanet partners [8]. All of the tools and technologies presented in this paper are open source, available at the Quintelligence GitHub repository [11].

### 3 Using Streamstory to explore Influenzanet data

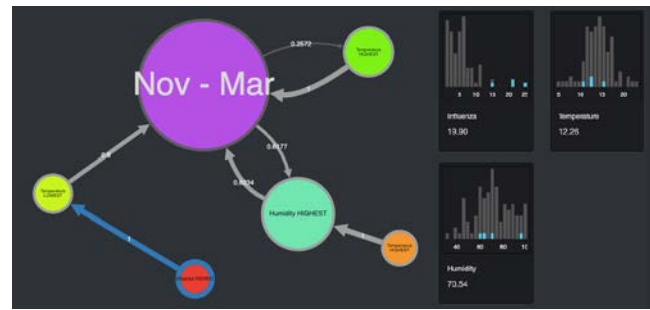
In the context of the visualization of complex data, the problem of visualization for the analysis and exploration of large multivariate time series is addressed by the Streamstory system [9][10]. This system computes and visualizes a hierarchical Markov chain model which captures the qualitative behaviour of the systems’ dynamics. It provides us with a multi-scale representation of the data based on a hierarchical model which allows us to interactively find suitable scales for interpreting the data.



**Figure 4: New data-driven seasons of the ILI incidence identified, subdividing the regular ILI season (marked as season 1, season 3, etc.), learning from historical Influenzanet data from Portugal during 2005-2013**

We consider Streamstory in the context of the MIDAS project to look into the seasonality definition of ILI based on the sourced data from the Influenzanet platform. This system was developed by the AI Lab at the IJS and refocused by Quintelligence within the MIDAS project to visually analyse the Influenzanet dataset. It is Open Source. In this research, we consider the data across 8 seasons for Portugal, from 2005 to 2013 to try to identify time intervals

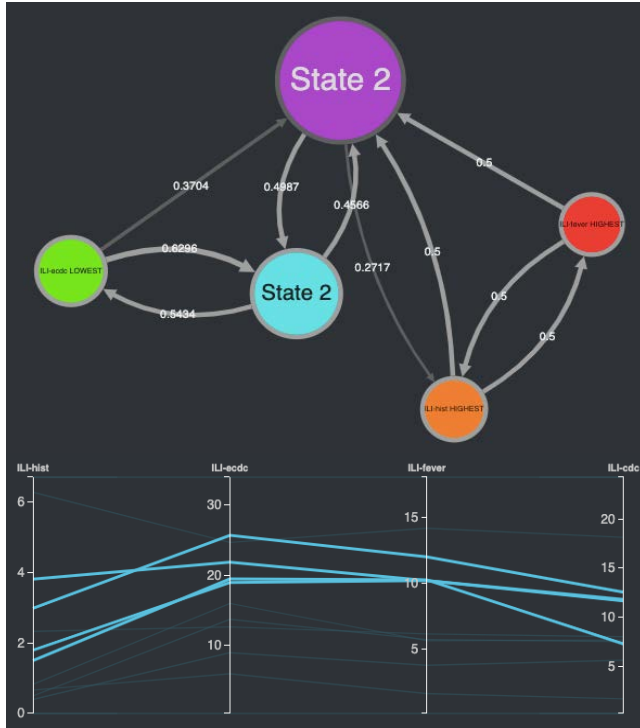
during the ILI season where the dynamics of the time-series behaves similarly. In this first analysis we call data-season to each state and try to identify the most prominent ones throughout the ILI season. We can identify five seasons where most of the time is spent in the first and in the last data-seasons. Moreover, the data-seasons 3 and 4 seem to be skipped at times with a direct passage from data-season 2 to data-season 5 (see Figure 4). In a second analysis we used Streamstory to examine the relationship between the ILI incidence and two different dimensions of weather data: temperature and humidity (naturally correlated to rainfall). The diagram in Figure 5 shows that the second largest state is assigned to high humidity, eventually corresponding to the also high ILI incidence. The highlighted red coloured state of high ILI incidence is strongly related to high humidity but also low temperature which seems to be pointing to the weather in the end of winter.



**Figure 5: Correlation between the incidence of ILI and the different dimensions of weather data**

A third analysis brings us to the comparison between the behaviour of ILI according to five coordinates, four of which corresponding to different ILI definitions: (i) historical, (ii) ECDC, (iii) including fever, and (iv) CDC. The diagram in Figure 6 shows the dynamics of the data when considering the 8 ILI seasons in Portugal altogether, highlighting the differences between the ILI definition used when counting incidence. The low incidence when considering ILI definition containing fever seems to have a strong expression in this analysis, followed by the high incidence of ILI defined using the historical and the ECDC definitions. The largest state is not related to any of the definitions in particular. When looking to its coordinates diagram, we can observe a higher influence of the ECDC definition, followed by the definition including fever. The CDC definition and the historical definition seem to have low weight in this largest state. A global analysis can be made through Streamstory to compare the ILI incidence in different countries. In the example of Figure 7 we compare five ILI seasons for Portugal and Italy. The close relation between the ILI behaviour in the two countries is usually similar between December and February, according to the diagram of states. Portugal and Italy tend to act distinctly in particular for the peaks of the epidemic, usually happening in Italy in November and February. Such a visualization might help us better understand the global behaviour of the epidemics throughout Europe, complementing the statistics provided by the Influenzanet platform itself.

Given the early stage of the Streamstory technology, it is difficult to have a clear global view on the obtained results and their meaning. Nevertheless, the approach seems to be promising, enabling a versatile analysis of the behaviour of the data through the diagrams of states, but also through the complementary coordinate diagram and the components histograms.

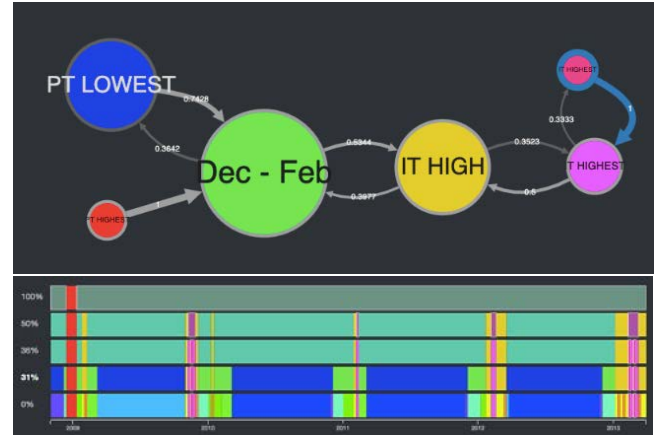


**Figure 6: Comparison of the behaviour of ILI across four ILI definitions (with states generated by ILI cases per week)**

## 4 Conclusions and further work

In this paper we discussed the research opportunities brought by the exploration of local data at each of the Influenzanet hubs. While the value of this data source is global, the local data explorations can target specific studies linked to priorities defined by the national public health institutes. The simplicity and ease of use of the discussed technology, offered as a service, permits the non-technical user to be much more independent of the in-house and overall technical support (often scarce in health organizations) to explore the local data in an almost real-time dimension. Moreover, some of the most meaningful data visualization models linked to those data explorations can be considered in the general platform to bring value to all the members of the Influenzanet network. Moreover, those visualization modules, common throughout the Influenzanet network, can provide means of comparison between countries and ILI seasons. Thus, the Kibana-based tool discussed in this paper can be of great value to digital epidemiology in general. Furthermore, the usage of advanced tools such as Streamstory might enable insights in the data that were otherwise unreachable. Versatile approaches such as this permit us to study

the behaviour of the data through its dynamics over a diagram of related states. In that, we can identify new data-driven seasons, relate the ILI season to the several coordinates of the weather data, and look through the weight of the different ILI definitions. Nevertheless, the meaningfulness of this kind of general approach demands a large effort on the interpretation of the results in the public health context that they belong to. Although the obtained results are good indicators to the promising potential of the usage of this technology, clear interpretations of the relations between states must be tackled within public health experts to enhance the usability of the technological tool as a public health tool.



**Figure 7: Comparison between the incidence of ILI in Portugal and Italy, and the corresponding state history**

## ACKNOWLEDGMENTS

We thank the support of the European Commission on the H2020 MIDAS project (GA nr. 727721).

## REFERENCES

- [1] C. Guerrisi et al (2016). Participatory Syndromic Surveillance of Influenza in Europe, *J Infect Dis.* 214 (supp 4): S386-S392
- [2] M. Hirsch, O. Woolley-Meza, D. Paolotti, A. Flahault, and P. Lukowicz (2018). gripeNET App: Enhancing Participatory Influenza Monitoring Through Mobile Phone Sensors. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (pp. 833-841). ACM.
- [3] MIDAS H2020 Project, "MIDAS Project Website," MIDAS Project, 2017. [Online]. Available: <http://www.midasproject.eu/>. [Accessed 8 8 2019]
- [4] S. P. van Noort et al (2015). Ten-year performance of influenzanet: Ili time series, risks, vaccine effects, and care-seeking behaviour. *Epidemics*, 13:28–36.
- [5] U.S. National Library of Medicine, "MEDLINE®: Description of the Database" [Online]. [www.nlm.nih.gov/bsd/medline.html/](http://www.nlm.nih.gov/bsd/medline.html/). [Accessed 2019]
- [6] D. Paolotti et al (2019). "Influenzanet 2.0. [Online] Available: [influenzanet.github.io/docs/](http://influenzanet.github.io/docs/)" [Accessed 20 8 2019]
- [7] D. Paolotti et al. Web-based participatory surveillance of infectious diseases: the influenzanet participatory surveillance experience. *Clinical Microbiology and Infection*, 20(1):17–21, 2014.
- [8] J. Pita Costa et al. (2018) "Influenzanet MIDAS toolset and demos" [Online]. [midas.quintelligence.com/midas-influenzanet-demos](http://midas.quintelligence.com/midas-influenzanet-demos) [Accessed 20 8 2019]
- [9] L. Stopar, "Streamstory," Institute Jozef Stefan, 2019. [Online]. Available: <http://streamstory.ijs.si/>. [Accessed 12 8 2019]
- [10] L. Stopar, P. Škraba, M. Grobelnik, and D. Mladenović (2018). StreamStory: Exploring Multivariate Time Series on Multiple Scales. *IEEE transactions on visualization and computer graphics* 25.4: 1788-1802.
- [11] Quintelligence, "GitHub repository" [Online]. Available: <http://github.com/quintelligence/>. [Accessed 20 8 2019]



# Feature Selection in Land-Cover Classification using EO-learn

Filip Koprivec  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
filip.koprivec@ijs.si

Jože Peternej  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
joze.peternej@ijs.si

Klemen Kenda  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
klemen.kenda@ijs.si

## ABSTRACT

Applying machine learning to Big Data can be a cumbersome task which requires a lot of computational power and memory. In this paper we present a feature selection technique for land-cover classification in earth observation scenario. The technique extends the state-of-the-art feature extractors by pruning the dimensionality of the required feature space and can achieve almost optimal results with 10-fold reduction of the number of features. The approach utilizes a genetic algorithm for generation of optimal feature vector candidates and multi-objective optimization techniques for candidate selection.

## Keywords

remote sensing, earth observation, machine learning, feature selection, classification

## 1. INTRODUCTION

Earth observation (EO) has become one of the major sources of Big Data. European Sentinel-2 mission, which acquires global data with 5-day revisit time, reports a total of 6.4 PB of satellite imagery products being available to the users via Copernicus services [2], whereas the total cumulative amount of EO data available from European Space Agency (ESA) is estimated to exceed 140 PB.

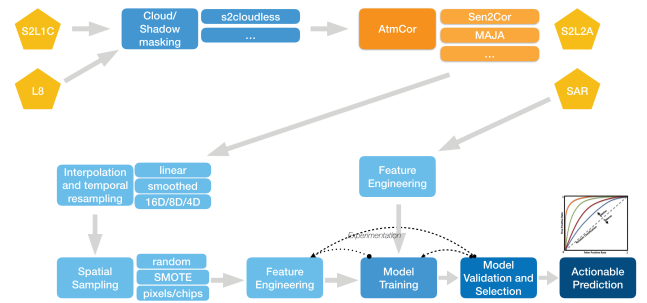
A huge amount of data have motivated EO and machine learning communities to invest into methodologies to work with such high volumes. Since 2016, as observed in Big Data from Space conferences, the community has tackled and solved the problem of storing, pre-processing and applications of basic machine learning and extensive deep learning algorithms for mainly solving classification problems. Processing pipelines have been established and are used regularly for solving different EO tasks [6].

The research has already approached the limits of the accuracy of the models. Our research has therefore focused on trade-off between model accuracy (of the current state-of-the-art) and processing efficiency. The approach is expected to be used in systems, which require a fast response with reasonably good results. Possible approaches include the use of fast classification techniques (i.e. Very Fast Decision Trees), which were taken from the field of stream mining, and optimization of the feature selection process.

This paper presents an early attempt to provide effective feature selection in land-cover classification. We illustrate that it is possible to significantly reduce the dimensionality of feature space of the state-of-the-art feature extractors [1, 8, 9] applied to a time-series of satellite images. Experimental data has been acquired by EO-learn library from PerceptiveSentinel<sup>1</sup> project.

## 2. DATA

Acquiring EO data is achieved using services provided by European Space Agency (ESA). For our experiments we have used Sentinel-2 missions data. This data includes scalar features from 13 different sensors with a resolution from 10 m × 10 m to 20 m × 20 m. A more detailed description of data available within Sentinel-2 missions is provided in [6]. EO-learn library [3] presents an abstraction layer over ESA services, which provide access and basic pre-processed (i.e. atmospheric correction, cloud detection and similar) products.



**Figure 1: Data flow (acquisition and pre-processing) with EO-learn library using Sentinel-2 data. EO-learn modules are depicted with light blue containers.**

Figure 1 depicts the data flow in a typical experiment. The top row depicts components for Level-1 and Level-2 pre-processing, which include cloud detection and atmospheric corrections. Products are being stored in the cloud and are accessed via EO-learn library. EO-learn modules are independent and can communicate with one another through a unified data structure (EO-patch) that can include satellite

<sup>1</sup><http://www.perceptivesentinel.eu/>

imagery data, enriched features, metadata and even corresponding vector data. For example: a feature engineering module for calculating normalized differential vegetation index (NDVI) from raw data would take **EO-patch** including the original 13 bands as an input and would output the same patch with an added NDVI index. Such modules are reusable and are being accumulated in the EO-learn library and made available to the community. Complex data processing and analytics pipelines can therefore be established literally within minutes.

### 3. METHODOLOGY

Based on satellite imagery our task is to classify land-cover in Slovenia. For this task we are using a time-series of images from the same year, which capture the dynamics of growth of particular vegetation and enable better accuracy of the models than a single image. Labels for building classification models have been acquired from a patch of land-use data (Slovenian LPIS data). The models can be applied to a wider area, where ground-truth data is not available and can even uncover some ground truth data mistakes (or generalizations). Our goal is to solve the task as fast as possible yet still accurate.

We base our methodology on the extraction of the state-of-the-art features from Sentinel-2 dataset. On top of this dataset we perform intelligent feature selection procedure based on multi-objective optimization approach.

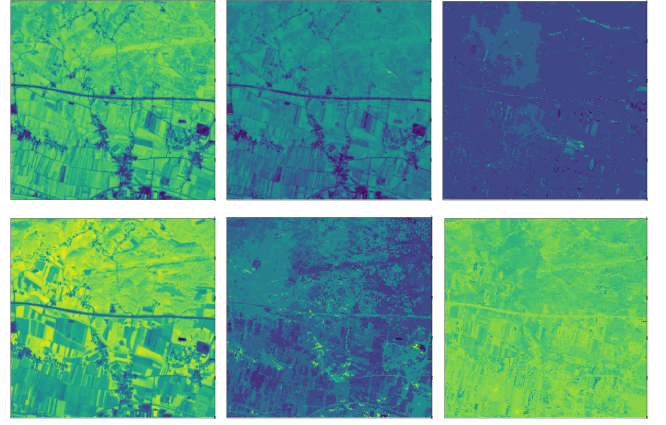
#### 3.1 Feature Engineering

We have acquired a time-series of satellite imagery for year 2017 and selected 27 small tiles ( $1 \text{ km} \times 1 \text{ km}$ ) from Slovenia randomly (ensuring, that appropriate distribution of different land-covers was consistent). We have performed cloud detection and then provided linear interpolation (simply because it is the fastest) over the remaining data points for each of the bands and additional indices. From these interpolated data we have extracted the phenological features suggested by Valero et al. [8]. The features have been calculated from following indices: NDVI, NDWI, EVI, SAVI, ARVI and  $\text{SPI}^2$  [5, 6]. These indices provide various information from the time-series which are important for land-cover classification (i.e. speed of growth, length of maximum index interval, etc.). All together we have used 108 different features within our experiments. Some examples of the features are depicted in Figure 2.

#### 3.2 Feature Selection

A feature selection algorithm should choose a limited amount of features out of the pool of 108, which would still provide enough information for almost optimal classification of land-cover. We employed a modification of the POSS genetic optimization algorithm [7] for the task. The algorithm would select a candidate solution (a selection of features) and slightly modify (mutate) it. The mutations have to be considered carefully, since the number of selected features must be kept as small as possible. The problem can be formulated as  $f : 2^N \rightarrow \mathbb{R}$ , where  $N$  is the number of all

<sup>2</sup>normalized differential vegetation index, normalized differential water index, extended vegetation index, soil-adjusted vegetation index, atmospherically resistant vegetation index and standardized precipitation index



**Figure 2: A sample of features extracted from a time-series of images: standard deviation of NDVI, max difference in NDVI in a sliding window, length of time interval where max mean value is attained (with specified tolerance), mean NDVI and rate of NDVI time-series change corresponding to the longest positive interval.**

features. We are looking for a subset  $A \subseteq 2^N$  that optimizes (minimizes or maximizes) the selected criterion function.

A naïve genetic algorithm without proper weighting of a number of features behaves poorly on most tested classifiers. If optimizing only the accuracy score (i.e.  $F_1$ ), the algorithm would almost always converge towards selecting all the features (since the dataset is large and there is generally no danger of overfitting). We modified the POSS algorithm to search possible feature space and optimize the number of selected features as well as the accuracy score with a 2-dimensional multi-objective optimization.

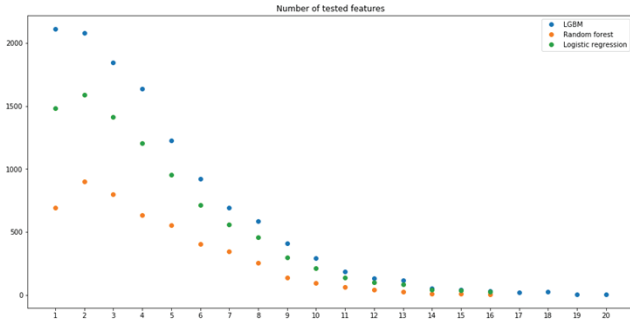
The main idea of the algorithm is as follows. We have  $N$  features, which we encode into a solution candidate  $S = \{f_1, f_2, \dots, f_N\}$ . A bit  $f_i$  represents whether the  $i$ -th feature is selected (value 1) in the candidate solution or not (value 0). We keep the current optimal elements on a 2-dimensional Pareto front, which is determined by  $1 - F_1$  score and number of selected features (for illustration see Figure 4). This approach can easily be extended to any other fixed dimension.  $1 - F_1$  is selected for convenience in selection (elements on Pareto front are those that are not comparable to any others in the current Pareto front, as determined by strict product order for each dimension, strict or non-strict is just a matter of preference when considering equality, but non-strict version more naturally excludes duplicates). In each iteration, the algorithm uniformly samples an item from the Pareto front and tries to improve it. Each bit  $f_i$  is then flopped with probability  $\frac{1}{N}$ , where  $N$  is the number of features.

This newly constructed candidate is then evaluated for its performance ( $F_1$ ). All the items on the Pareto front are then compared with this new item. If there exists no such item that is comparable or bigger from the new item, the new item is on the Pareto front and is subsequently added to it. All items that are comparable or smaller than new item

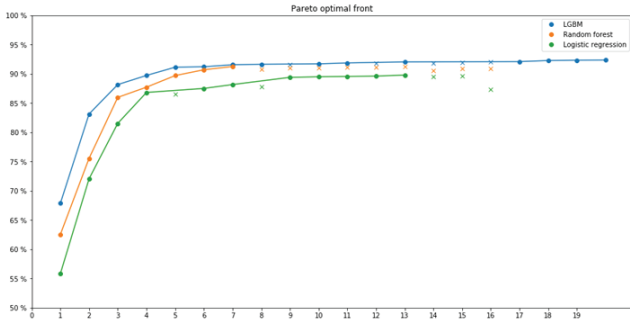
are removed from the Pareto front, as they are (strictly) Pareto sub-optimal. Strictness is useful since it removes the duplicates (in a non-strict product weak ordering, even if the relation is non-linear, as in the case in the Pareto front, the product ordering is antisymmetric) [5].

## 4. RESULTS

Results of the early feature selection experiments are depicted in Figures 3 and 4. We have tested the methodology with the most popular classification techniques used in remote sensing (apart from deep learning): gradient boosting (LightGBM implementation [4]), random forests and logistic regression (baseline). Gradient boosting has proven to be a superior method whereas logistic regression performed the worst. SVM classifier was not considered since its training time complexity  $\mathcal{O}(N^3)$  is too high for frequent re-training, needed in the feature selection algorithm.



**Figure 3: Number of tested candidates ( $y$ ) per number of features ( $x$ ). Gradient boosting is depicted with blue, random forests with orange and logistic regression with green dots.**



**Figure 4: The best candidate  $F_1$  score ( $y$ ) per number of features ( $x$ ). The lines depict the Pareto front for a particular classification algorithm. The optimal number of features for random forests and logistic regression is smaller than the size of the longest tested feature vector. Gradient boosting is depicted with blue, random forests with orange and logistic regression with green colour.**

Figure 3 depicts the number of tested candidates per number of features. Number of features starts to decline sharply, but is a bit jumpy. This represents an expected behaviour considering the random nature of the feature selection algorithm and incremental difficulty of greatly increasing the number of features.

Figure 4 shows that smaller number of tested examples with a high number of features does not significantly affect  $F_1$  score (considering small changes of a random element on the Pareto front, this seems reasonable). The same figure also shows, that already with a careful selection of just a few "good" features, classification produces quite good results. The figure also nicely depicts part of Pareto front and shows that high quality of feature selection might also improve the classification in some cases.

A clear plateau shape can be seen in Figure 4, hinting, that there is a reasonable choice of a subset of features. Selecting a small, but optimal subset of all features can yield good accuracy score of the classification algorithm, with decreased memory and computation footprint. The most important consequence of using an optimal subset of features is, that it saves a lot of time for data preparation (not extracting unneeded features, not sending/saving unneeded data) and most importantly makes the model reasonably small and fast, which allows usage even on a plethora of low computational power devices.

In the results presented above, LightGBM classification algorithm performance is unmatched by either random forest or logistic regression. This is an expected result since boosting can skew the feature space and can inherently introduce non-linear features into the model. The most illustrative case for the strength of proper feature selection is however seen in the case of random forest algorithm. We can observe from Figure 4 that already with 7 wisely chosen features (out of 108) one can achieve the optimal  $F_1$  classification score. The reduced number of features speeds up the feature extraction step (less features need to be calculated) and modeling (less data is needed, fewer features are considered) and reduces the memory consumption demand.

## 5. CONCLUSIONS AND FUTURE WORK

This is the early paper on feature selection used for land-cover classification. It shows great potential of the methodology and up to 15-fold reduction of the number of needed phenological features in order to still achieve state-of-the-art accuracy. The methodology could be used with potentially great benefits also on other types of feature vectors in land-cover classification (i.e. with resampled index values), where it would automatically find the features that can distinguish between various land-cover classes. The main underlying reason for our research lies in the provision of computationally effective methods for faster, easier and cheaper EO data analysis.

There are still research challenges to be considered in this work. Firstly, benefits of feature reduction to the computational tasks should be examined in depth. The most important phase of the process is the inference phase (land-cover classification on large areas). However, preliminary results indicate that speed-up and memory consumption might be smaller than expected based on common sense.

Feature selection should be tested with other faster classification methods (i.e. based on incremental learning [5]), which trade accuracy for the faster computation. The latter might be beneficial in particular use cases (i.e. on-the-fly classification for on-line EO browsers like SentinelHub or

large scale classification). A comprehensive study of benefits within full-stack pipelines (from data acquisition to inference) should be conducted.

Earth observation community has striven towards achieving optimal accuracy of the classification algorithms in the past few years. Especially deep learning algorithms have shown to require vast amounts of computational time, which is sometimes difficult to obtain. Presented work, together with research into computationally effective classification methods, might be a step towards sacrificing some of the accuracy in order to achieve final results sooner and with less struggle.

## 6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project PerceptiveSentinel (H2020-EO-776115). The authors would like to thank Sinergise for their contribution to EO-learn library along with all help with data analysis.

## 7. REFERENCES

- [1] BELGIU, M., AND CSILLIK, O. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sensing of Environment* 204 (2018), 509 – 523.
- [2] EUROPEAN SPACE AGENCY. Mission status report 152. <https://sentinel.esa.int/documents/247904/3720568/Sentinel-2-Mission-Status-Report-152-25-May-28-June-2019.pdf>. Accessed: 2019-08-01.
- [3] H2020 PERCEPTIVESENTINEL PROJECT. Eo-learn library. <https://github.com/sentinel-hub/eo-learn>. Accessed: 2019-09-06.
- [4] KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., YE, Q., AND LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (2017), pp. 3146–3154.
- [5] KOPRIVEC, F., KENDA, K., ČERIN, M., BOGATAJ, M., AND PETERNELJ, J. H2020 Perceptive Sentinel - Deliverable 4.6 Stream Mining Models for Earth Observation. Reported 31st March 2019.
- [6] KOPRIVEC, F., ČERIN, M., AND KENDA, K. Crop Classification using Perceptive Sentinel. In *Proc. 21th International Multiconference* (Ljubljana, Slovenia, 2018), vol. C, Institut "Jožef Stefan", Ljubljana, pp. 37–40.
- [7] QIAN, C., YU, Y., AND ZHOU, Z.-H. Subset selection by pareto optimization. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1774–1782.
- [8] VALERO, S., MORIN, D., INGLADA, J., SEPULCRE, G., ARIAS, M., HAGOLLE, O., DEDIEU, G., BONTEMPS, S., DEFOURNY, P., AND KOETZ, B. Production of a dynamic cropland mask by processing remote sensing image series at high temporal and spatial resolutions. *Remote Sensing* 8(1) (2016), 55.
- [9] WALDNER, F., CANTO, G. S., AND DEFOURNY, P. Automated annual cropland mapping using knowledge-based temporal features. *ISPRS Journal of Photogrammetry and Remote Sensing* 110 (2015), 1 – 13.

# Identifying events in mobility data

Branko Kavšek<sup>1,2</sup>

<sup>1</sup>Artificial Intelligence  
Laboratory

Jožef Stefan Institute  
Ljubljana, Slovenia  
branko.kavsek@ijs.si

<sup>2</sup>Department of Information  
Sciences and Technologies  
University of Primorska  
Koper, Slovenia  
branko.kavsek@upr.si

Dunja Mladenec

Artificial Intelligence  
Laboratory

Jožef Stefan Institute and  
Jozef Stefan International  
Postgraduate School  
Ljubljana, Slovenia  
dunja.mladenec@ijs.si

Omar Malik

Network Science and  
Technology Center  
Rensselaer Polytechnic  
Institute  
Troy, USA  
maliko@rpi.edu

Boleslaw K. Szymanski

Network Science and  
Technology Center  
Rensselaer Polytechnic  
Institute  
Troy, USA  
szymab@rpi.edu

## ABSTRACT

Today we are used to being interconnected via our smartphones and having our phone location tracked by different apps. ICT technology enables real-time monitoring and processing the user location data from GPS coordinates of a phone. Based on observing the user mobility, Artificial Intelligence methods can be used to improve transportation, proactively provide mobility recommendations and acquire knowledge using the user context. This paper describes the application of machine learning algorithms on user mobility data to identify and understand potentially interesting events. The data for this research was collected from a sample of users consenting to be monitored through our in-house developed smart phone app. A pilot study that includes 227 users that were tracked over a period of 7 years yields fairly positive evaluation results in terms of predictive accuracy of identified events but succeeds in identifying exclusively “well-known” events related to users going to or coming from the office and/or lunch. This shows that machine learning methods can be a suitable choice for identifying events in mobility data but there is still room for improvement.

## CCS CONCEPTS

• CCS Information systems Information systems applications Data mining

## KEYWORDS

Users mobility, network analysis, event detection, machine learning, clustering.

## 1 INTRODUCTION

Given the data of user mobility, we were looking into using social network analysis and machine learning methods to understand causal templates and identify and predict events in the user mobility data. To this end we have defined an event as an action that is a consequence of some user and/or environment property. For instance, such event is the user driving in the morning if the weather is cold, otherwise the user would be using some other means of transportation. The weather being cold is a cause for the event of driving.

The idea for identifying events is to build a social network of locations that the users are frequently visiting and compare traces of different users to identify typical behaviors. Once we have the traces of typical behaviors, we look for significant diversions in traces and hypothesize that they are consequences of some specific user or environmental context, for instance, from work the user is usually going home but every Tuesday afternoon we observe that the user is going to gym instead not to home. We use machine learning methods to categorize the events based on identified properties of the users/environment correlated with diversions of traces (these properties are seen as potential cause of an event). For instance, on Tuesday afternoons, when the previous location is work, the user frequently uses a bicycle. Then we find regularities in the properties to group the events (and causes). For instance, under specific circumstances some users go from work to gym instead of going home (relevant circumstances here could be that a user likes exercising and the period is Tuesday afternoon).

This paper is organized as follows. Section 2 shortly lists all related research that was done on the same or very similar data to the data used in this research. In Section 3 the data is presented together with the performed pre-processing. Section 4 describes the experimental evaluation with descriptions of the methodology and results. Section 5 provides interpretation and discussion of the experimental results. Section 6 concludes the paper and gives directions for future work.

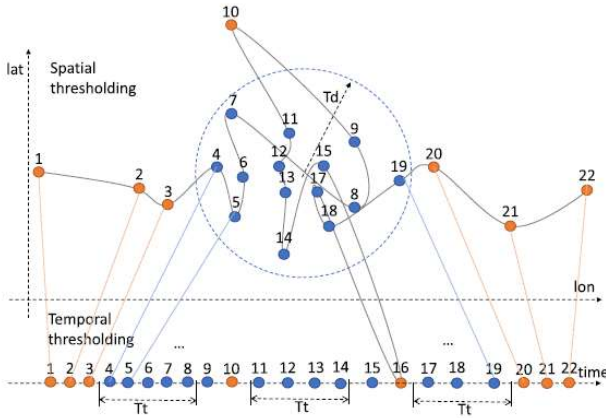
## 2 RELATED RESEARCH

When referring to user mobility data nowadays, we mostly refer to GPS data provided by the user’s smart phone or other wearable device. Sometimes, this data also includes the readings of other sensors, if present and functional (e.g. accelerometer). Tracking a user thus means collecting a series of GPS coordinates readings in a time sequence.

In [6] the authors argue that the raw GPS data is noisy and messy. Thus, when analyzing user paths, they group the GPS coordinates based on time and distance resulting in the detection of the so-called stay-points or locations where a user spent more time. Figure 1 depicts the idea of stay-point detection by clustering in



space and time. The blue points in Figure 1 represent a spatio-temporal cluster of GPS coordinates called a stay-point.



**Figure 1: Outlier removal and spatio-temporal clustering [6]**

The authors of [7] went a step further and used the data produced by the previous spatio-temporal clustering [6] to try to identify and give a rating to points of interest (PoI) based on users' behavior.

In [2] natural language processing (NLP) methods are used by the authors in combination with previously mentioned methods and crowdsourcing to provide additional user context.

Predicting users' mobility is what the authors of [5] tried to achieve by predicting the next location and mobility pattern of a user using probabilities and the Markov state-space model.

In [9] the authors aimed at detecting the most likely transportation mode of a user and in [1] they tried to motivate the users to make a more ecologically-friendly transportation choices.

Finally, the authors of [8] devised a methodology for visualizing qualitative patterns in multivariate time series that was tested also on user mobility data.

### 3 EXPERIMENTAL DATA

Raw data was collected from 432 users in a span of nearly 7 years (from 5.7.2012 until 7.4.2019). The users installed a mobile app that tracked their whereabouts by sending a reading to the database every 30 seconds. Every reading sent to the database included: the "activity ID", "the user ID", "timestamp", "GPS coordinates" (LAT and LON), additional data (accelerometer readings, GPS accuracy readings, ...). Not all 432 users were sending data continuously for all 7 years (some users came later or left earlier, some smart phones switched off because of power source issues, sometimes GPS signal was out of range, ...).

Because GPS data from users' phones was noisy and messy as argued by the authors of [6], we used their method to preprocess our raw data. The pre-processing steps taken to "clean" our data are described in Section 3.1.

#### 3.1 Pre-processing the data

Data pre-processing was performed in two steps. First, clustering in space and time was applied to a set of uninterrupted

30 seconds GPS readings. Second, any remaining outliers were removed.

##### 3.1.1 Clustering in space and time

This type of clustering is best understood by looking at Figure 1 (taken from [6]). Points, marked with numbers from 1 to 22 and connected with a line in this figure, represent a series of 22 uninterrupted 30 seconds GPS readings from one user. Every point has an associated timestamp and the values for LAT and LON. The clustering is performed using one time and one space threshold. A time threshold of 5 minutes and a space threshold of 120 meters (the threshold values that were actually used throughout our experiments) mean that all GPS readings that fall within a radius of 120 meters for more than 5 minutes will be clustered together to form one stay-point. Start and end times in this stay-point correspond to the first and last GPS readings in the cluster, respectively. A GPS coordinate for this stay-point is the average of LAT and LON values of all GPS readings in the cluster. The non-clustered GPS coordinates represent the so-called paths. In Figure 1 we can notice two paths (1-3 and 20-22) and 1 stay-point (all blue points 4-19).

##### 3.1.2 Outlier removal

When performing the spatio-temporal clustering described in Section 3.1.1, we requested that all the remaining paths must contain at least two GPS coordinates. The GPS coordinates that do not belong neither to a stay-point, nor to a path after clustering, are considered outliers and thus removed.

##### 3.1.3 The pre-processed data

After clustering and outlier removal described in Sections 3.1.1 and 3.1.2, the data contains 235,683 records, of which 114,923 are stay-points and 120,760 are paths. Every stay-point is described by a start time, an end time and a GPS location of its center. The paths, on the other hand, are ordered sets of readings, where each reading has a timestamp and a GPS location. Some paths can contain hundreds of readings, some of them can even be circular (starting and ending in the same GPS location).

Since some of the users that were tracked traveled a lot to all parts on the globe, we decided to simplify things by considering just those stay-points and paths for which all GPS coordinates were inside a rectangle (N 45° - 47° LAT, E 13° - 17° LON) that is limited to Slovenia in the Ljubljana nearby area. This also simplified our dealing with time, as all the data is in the same time zone. We also did not consider daylight-saving times. This reduction leaves our data with 110,072 records from 227 users, of which 58,188 are stay-points and 51,884 are paths.

Since our goal is to identify events in user mobility data, we need additional features describing the data that may later serve as event descriptors. The only two features we have at the moment are "time" and "position" (in space). From "time" we created six new features as follows:

- Time of day,
- Hour,
- Weekday,
- Weekend,

- Season, and
- Holiday.

“Time of day” is a discrete feature with 10 values (see Table 1), “Hour” is just the hour part of the timestamp, “Weekday” is a discrete feature with values MON – SUN, “Weekend” is a binary feature (T if SAT or SUN, F otherwise), “Season” is one of the 4 seasons (Winter, Spring, Summer or Autumn), Holiday is a discrete feature denoting all known Slovenian holidays.

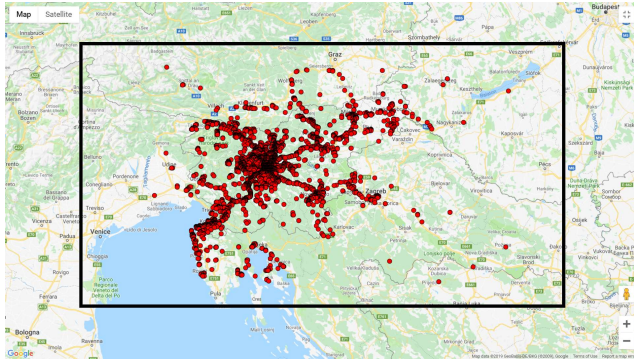
**Table 1: Values for the discrete feature “Time of day”**

<i>Timestamp</i>	<i>Value</i>
6 AM – 8 AM	Early morning
8 AM – 11 AM	Morning
11 AM – 1 PM	Mid-day
1 PM – 3 PM	Early afternoon
3 PM – 5 PM	Afternoon
5 PM – 7 PM	Late afternoon
7 PM – 10 PM	Evening
10 PM – 12 PM	Late evening
12 PM – 4 AM	Night
4 AM – 6 AM	Dawn

From “position” we created just one additional feature, namely “Region” that maps a GPS coordinate to one of the 5 geographic regions in Slovenia (Štajerska-Premurje, Dolenjska-Hrvaška, Primorska-Istra, Gorenjska-Avstrija, Gorica-Italija); a sixth “region” was added for the capital (Ljubljana).

## 4 EXPERIMENTAL EVALUATION

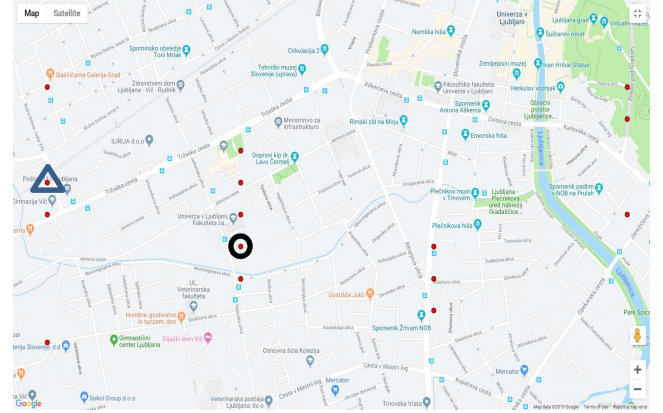
In this experiment we decided to additionally simplify things by “reducing” all the 51,884 paths to just the initial and final positions, disregarding all the 30-seconds position readings in-between. By doing so the notions of “path” and “stay-point” lose their meaning, since now we can consider a stay-point as a path whose initial and final positions are the same. Thus, we can drop the “type” (stay-point/path) feature and consider all 110,072 activities from 227 users in the same way.



**Figure 2: Visualization of experimental data on the map of Slovenia (Google Maps API)**

We also decided to round all LAT and LON values to 2 decimal places. This was done since minor fluctuations in the GPS signal were being treated as different locations. By reducing our precision, we smoothed out this noise. The visualization of this data on a map of Slovenia is shown in Figure 2 – the black rectangle represents the observed region.

We now observe the 20 most visited GPS locations. 18 of the 20 most visited locations are all located around or near one of the most popular locations – they are depicted in Figure 3, with the black circle representing the most popular one. For each location we sample all paths that contain this location either at the beginning, the end or on both sides. This generates 20 new datasets. Just the results for the dataset associated with the most frequent location is presented in this paper, since for the other 19 datasets the results are very similar, and this is just the first experiment intended to be more of a proof of concept than a thorough result.



**Figure 3: The 18 most popular locations (Google Maps API)**

The most frequent location’s dataset now contains 17,582 activities (2-point paths). At this point we decide to observe the difference between users that come to the most frequent location, those that leave the most frequent location and those that stay at the location. We create a new *Class* attribute that will serve as our dependent variable for the predictions and assign the values “In” (5,356 examples), “Out” (6,947 examples) and “Stay” (5,225 examples) to it, reflecting the users coming, leaving or staying. So, we end up with a dataset with 17,582 examples, 14 independent attributes – (6 for “time”, 1 for “space”) x 2 (for start and end point) and a quite balanced class attribute.

### 4.1 Methodology

For machine learning we used the WEKA workbench [3,4]. The algorithms used were PART (rule learning), J4.8 (decision trees), SMO (SVM), Random Forrest and Naïve Bayes. All the algorithms were ran with the default parameters; the evaluation was performed using 10-fold cross-validation observing classification accuracy as the performance measure. The task we are addressing is supervised learning to build a model for distinguishing between the three types of users that are visiting the most frequent location. In our data the most frequent location turned out to be the Jožef Stefan Institute, which is the working place for most of the users.

## 4.2 Results

The results of this experiment are presented in Table 2. Classification accuracies are presented as percentages together with standard deviations.

**Table 2: Results of selected algorithms on most frequent location’s dataset**

<i>ML algorithm</i>	<i>Average accuracy (%) / STD</i>
<i>Majority class</i>	39.5 (“Out”)
Naïve Bayes	65.5 / 2.45
J4.8	68.1 / 2.76
PART	68.5 / 2.19
<b>SMO</b>	<b>71.4 / 1.99</b>
Random Forrest	70.2 / 2.01

The results in Table 2 show that all five algorithms perform within the 65% to 70% classification accuracy, with SMO having slightly higher accuracy. The majority class value in this case is “Out” appearing in just 39.5% of all the examples.

Not shown in Table 2 is the co-occurrence of certain attribute values with specific class values: “Time of the day = Morning” frequently co-occurs with class value “In” in the generated models; “Time of the day = Mid-day” frequently co-occurs with both class values “In” and “Out”; “Time of the day = Late afternoon” frequently co-occurs with class value “Out”. There is a lot of migration between the most frequent location (black circle on Figure 3) and one of the other top 20 frequent locations (blue triangle on Figure 3).

## 5 DISCUSSION

As the results in Table 2 clearly show, the Support Vector Machine classifier (SMO) has the highest accuracy, but the difference compared to the second best, Random Forrest, is not big. All selected machine learning algorithms clearly outperform the majority classifier with around 70% accuracy.

The frequent co-occurrence of attribute values with specific classes show the following:

- in the morning people tend to come “In” to the frequent location (they come to work),
- in the late afternoon people tend to go “Out” from the frequent location (they leave the office),
- at mid-day (around noon), both “In” and “Out” links suggest people go for lunch or a snack,
- a lot of migration between the most frequent location and one of the other frequent locations suggests people have some sort of engagement on this other frequent location – indeed it turned out that the other frequent location is in fact the building where a lot of mobility users work in their spin-off companies.

In Figure 3 the “grid effect” of rounding up the GPS coordinates is clearly visible and sometimes the rounded coordinates do not correspond exactly to the physical locations of the points-of-interest.

## 6 CONCLUSIONS AND FUTURE WORK

Our pilot study on identifying events in mobility data provided fairly positive experimental evaluation results in terms of predictive accuracy of identified events. However, the events identified are “well-known” events related to the users going to or coming from the office and/or lunch.

On the other hand, over-simplification of the mobility data did not “pay off” in our case, which is clearly visible in the form of the “grid effect” of rounded GPS positions and lack of interesting/surprising relationships in the constructed models.

One possible direction that we are looking at for the future research is to re-run the experiments on the original pre-processed data (described in Section 3) and focus our attention on the changes in user paths. Instead of rounding the GPS coordinates changing the parameters of the clustering used for stay-point creation described in section 3.1.1 seems to be a more promising direction to take.

## ACKNOWLEDGMENTS

This work was partially supported by the Slovenian Research Agency and the European Commission RENOIR project H2020-MSCA-RISE-691152.

## REFERENCES

- [1] E. Anagnostopoulou, J. Urbančič, E. Bothos, B. Magoutas, L. Bradeško, J. Schrammel, G. Mentzas. *From mobility patterns to behavioural change: leveraging travel behaviour and personality profiles to nudge for sustainable transportation*. Journal of Intelligent Information Systems, pp. 1-22, 2018.
- [2] L. Bradeško, M. Witbrock, J. Starc, Z. Herga, M. Grobelnik, D. Mladenčić. *Curious Cat-Mobile, Context-Aware Conversational Crowdsourcing Knowledge Acquisition*. ACM Transactions on Information Systems (TOIS) 35 (4), 33, 2017.
- [3] E. Frank, M. A. Hall, I. H. Witten. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, Fourth Edition, 2016.
- [4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Volume 11, Issue 1, 2009.
- [5] B. Kažič, J. Rupnik, P. Škraba, L. Bradeško, D. Mladenčić. *Predicting Users’ Mobility Using Monte Carlo Simulations*. IEEE Access, 2017.
- [6] M. Senožetnik, L. Bradeško, B. Kažič, D. Mladenčić, T. Šubic. *Spatio-temporal clustering methods*. Conference on Data Mining and Data Warehouses (SiKDD 2016).
- [7] M. Senožetnik, L. Bradeško, T. Šubic, Z. Herga, J. Urbančič, P. Škraba, D. Mladenčić. *Estimating point-of-interest rating based on visitors geospatial behaviour*. Computer Science and Information Systems 16 (1):131-154, 2019.
- [8] L. Stopar, P. Škraba, M. Grobelnik. *Streamstory: exploring multivariate time series on multiple scales*. IEEE Transaction on Visualization and Computer Graphics, 12(8):1-10, 2018.
- [9] J. Urbančič, L. Bradeško, M. Senožetnik. *Near real-time transportation mode detection based on accelerometer readings*. Proceedings of the 19th international multiconference information society (IS 2016).



# Early land cover classification with Sentinel 2 satellite images and temperature data

Matej Čerin  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
matej.cerin@ijs.si

Filip Koprivec  
Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana,  
Slovenia  
filip.koprivec@ijs.si

Klemen Kenda  
Jožef Stefan Institute  
Jožef Stefan International  
Postgraduate School  
Jamova 39, 1000 Ljubljana,  
Slovenia  
klemen.kenda@ijs.si

## ABSTRACT

The weather is one of the main factors for events that happen on the surface of the earth. Surprisingly, no effort was made to use weather features together with satellite images for land cover classification. In the paper, we use temperature data along with satellite images to improve the accuracy of the classification and to get classification as early in the year as possible. Every year has different conditions, so the temperature can be used as an objective criterion at what time to do classification.

## Keywords

remote sensing, weather, earth observation, machine learning, feature selection, classification, temperature

## 1. INTRODUCTION

Precise classification of land cover is important for agriculture and food security. It is especially important to make the classification as early as possible so the farmers can know the situation of their land better and can take appropriate action.

Since ESA launched the Sentinel 2 mission that provides big amount of data, a lot of research is focusing on satellite data and land cover classification [9, 5, 15, 14]. We found no paper where the weather would be used to improve the land cover classification. Most of the research using weather is focused on predicting the yield [11, 6, 1].

Literature shows that weather plays one of the most important roles in the growth of vegetation and development of crops [8, 7, 16]. It is expected that it will play an even more important role in the future, with climate change trends [7]. According to literature the most important weather variables for the development of vegetation are temperature, precipitation and the duration of sunlight. The most important of them is temperature. The research shows that time for growing is better correlated with temperature than the number of days from planting [1]. The plant growing models talk about the fact that the plants have some optimal temperature above which the plants grow best. It is also important that the temperature does not go above some threshold [8]. Weather variables are also important for farmers decisions. Based on weather condition the farmer can choose when what and how to seed [7].

In the paper, we try to improve our classification prediction with the help of temperature. We are trying to find the moment when the temperature is above some threshold for long enough.

## 2. DATA

### 2.1 Data Acquisition

In the article, we use satellite data from the ESA Sentinel-2 mission [3]. The Sentinel-2 mission has two satellites that circle the earth with 180° phase. The same point is visited at least once every five days. Satellites collect data in 13 different spectral bands. Spatial resolution is 10m, 20m or 60m, depending on the band.

We downloaded satellite data with the sentinel-hub library [12] integrated in the eo-learn library [13]. Eo-learn is the library that makes access to and processing of earth observation data easier. We used eo-learn also to preprocess the data. Data were downloaded for times between July 1, 2015 and June 30, 2018.

Temperature data are from the ECMWF (European Centre for Medium-Range Weather Forecasts) archive [2]. ECMWF is the archive that stores accurate historical data, both observed and forecasts weather data for the world. The data used in the article has approximately  $15 \times 15$  km resolution.

Data for land cover are from the website of the Slovenian Ministry for farming, forests, and food [10]. Data are publicly available and contain 25 classes. The land use data are mostly created with aerial photography called orthophoto. In case that the area is impossible to categorize from the image, the terrain inspection is made.

### 2.2 Data Preprocessing

Most preprocessing of satellite data are already made by ESA, like atmospheric reluctance or projection [4]. Thus, our data is already clean and ready for use.

The biggest challenge in our satellite data set was missing data when the clouds cover some images or some parts of it. To eliminate that problem we took images provided by ESA and filtered out the pixels that were covered by clouds using the cloud mask. The cloud mask was provided by eo-learn's AddCloudMaskTask() task. That way we got images only with cloudless pixels.

The other concern was that all images were not taken on the same date and now we have some missing data from cloud removal. Therefore we took a time series of each pixel and linearly resampled all bands over time. We resampled it on every 16th day starting on 1.1.2016 and up to 31.12.2017.

That way we produced a data set that had all images at the same timestamp. Linear resampling also filled the gaps from filtering the images with clouds.

The land cover data had 26 different classes, but some classes were too small. We joined the related classes under five more general classes (grass, forest, crop land, urban area and other).

### 3. METHODOLOGY

The idea of the experiment was that when the temperature is above a certain threshold the plant starts to grow. Because they grow differently, it is easier to classify the areas with different vegetation. Therefore we looked for the time when the temperature is high enough for land covers to be easier to classify.

#### 3.1 Feature Vectors

Experiments were conducted in the area of Slovenia. We used data from years 2016 and 2017. Data from 2016 were used to train the model and data from 2017 for testing. We randomly chose 150 patches in the size of  $50 \times 50$  pixels ( $500 \text{ m} \times 500 \text{ m}$ ). Then we sampled from those patches approximately 50 000 pixels with the class that we are interested in and 50 000 pixels that are not from that class. That way we get balanced data sets, that we can use to train our learning algorithm. Thus, we created two vectors for each class, One for learning and one for testing.

For each date, we counted the number of days that average temperature exceeded some maximum temperature ( $T_{max}$ ). We calculated that for  $T_{max}$  from  $-10^{\circ}\text{C}$  to  $26^{\circ}\text{C}$  for every  $2^{\circ}\text{C}$ . Those features we added to the time series of pixels. Because the temperature data has smaller spatial resolution than satellite data, we appended to each pixel the temperature data from the weather data point that is the closest to the coordinates of that pixel.

For all pixels' time series, we found the first timestamp for which the number of days with temperature above  $T_{max}$  was higher than the chosen number of days. We took the values of bands at that timestamp for each pixel. That was done for both years, resulting in two feature vectors, one to train the model and the other to test it.

Because some higher temperatures were not reached often enough, some did not include all pixels. If less than 70% of all pixels passed the criteria, the experiment under those criteria was not made.

#### 3.2 Experiment

On data sets from 2016, we trained the decision tree classifier. The decision tree function is from the sci-kit learn python library and was used with default settings. To evaluate models we calculated predictions for the year 2017, and calculated  $F1$  score. In all experiments we made two class classification.

We did experiments systematically for all calculated temperatures  $T_{max}$  and for all possible numbers of days from 1 to 30.

To compare results we made another experiment where we trained the model on the data from one date and tested it on the closest date next year. We compared  $F1$  scores from both experiments, to see if the model, trained with data set chosen with help of temperature, perform better.

## 4. RESULTS

The maximum  $F1$  score from the experiment with the data set determined by temperature is better for all classes than the maximum  $F1$  score from the second experiment (table 1). That means that the temperature helped us improve the classification.



Figure 1:  $F1$  scores from model trained and tested at the same date. Maximum values are used to compare results from first experiment.

	Forest	Grass	Crop	Urban area	Other
Max $F1$ score-temp	0.76	0.74	0.70	0.73	0.59
Max $F1$ score-time	0.74	0.67	0.62	0.64	0.53
Diff	0.02	0.07	0.08	0.09	0.06

Table 1: Table shows maximum  $F1$  scores for all five classes, from both experiments and the difference between them.

Figure 2 shows the difference between  $F1$  scores from the experiment with temperature and the maximum  $F1$  score from

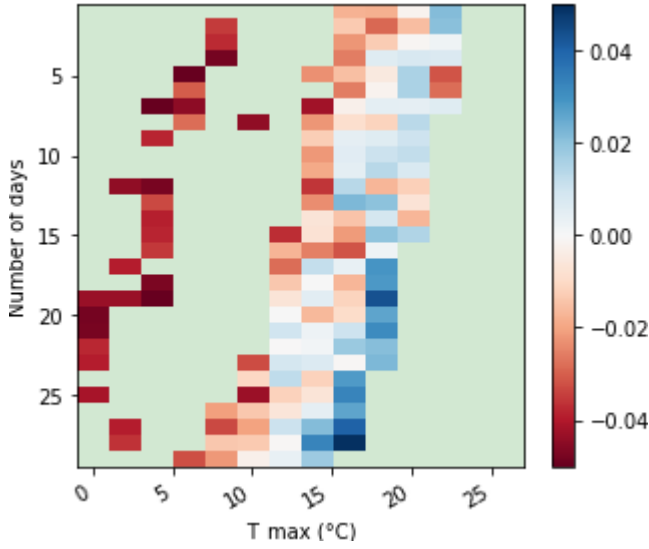


Figure 2: The figure shows the matrix of difference between  $F1$  scores from the experiment with temperature and the maximum  $F1$  score from the second experiment. Blue color shows times when the  $F1$  score is better than in first experiment and red color when the  $F1$  score is worse by less than 0.05. Green area is where the classification was worse by more than 0.05 or the times when less than 70% of pixels were available for training. This figure is from the classification of grass.

other experiment. The experiment from figure 2 was made for grass classification. We get similar images also for other classes. On the figure, we notice that we get two islands, one at the temperature around  $4^{\circ}C$  and the other around  $16^{\circ}C$ . Each island corresponds to data at different dates. The data from the same island are from similar dates. The distribution over dates for both islands is shown in figure 3.

The classification of grass, forest, and urban area produces the same kind of islands, while the crop and other produces only one big island. We assume that this is due to non-homogeneous vegetation in those two classes.

From figure 1 we see that the classification of the grass in the second experiment is the best at the end of August ( $F1 = 0.67$ ). An even better classification score (up to 0.07,  $F1 = 0.74$ ) can be achieved with most of the classification made before august (blue, orange and green bars from figure 3).

Another useful result from approach with temperature is that the classification can be made earlier in the development of plants. Relatively good classification ( $F1 = 0.63$ ) can be achieved by the end of April (red, purple, brown and pink bars from figure 3).

For forest and urban area, the first island gives also slightly worse classification but we can classify earlier. While the second island gives us better results at the approximately same time. The classification of both classes with one island is approximately at the same time, but it achieves better results.

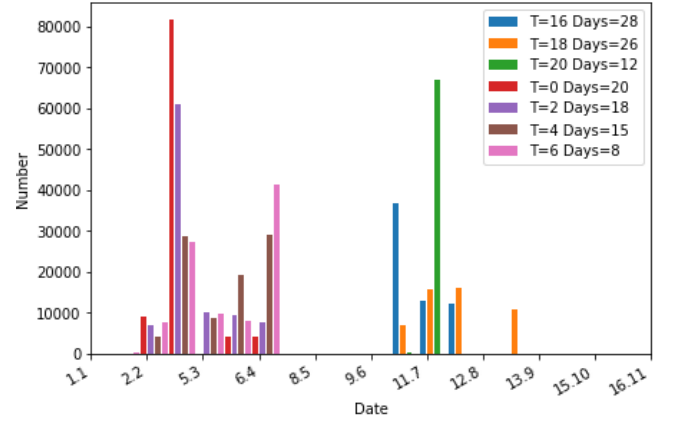


Figure 3: Histogram shows the number of pixels with certain time stamp, at chosen  $T_{max}$  and number of days. It shows only some representatives of both islands.

## 5. CONCLUSIONS

In the article, we showed that temperature can help us determine when is the most appropriate time to do classification. Because the conditions are not the same everywhere, temperature gives us a good and objective tool to determine when is the best time to do classification.

We also showed that we can do classification much earlier than we thought. Plants do not need to fully grow. We can identify its development as early as March or April.

The problem with that method is that we can not know in advance when the optimal time for classification will come. And when that time comes it is not the same for all areas but is determined locally, by local weather condition. Therefore usually, one model can do all classification in one month and a half, for the whole area of Slovenia. But for a farmer who is interested in the growth and development of his plants, that is usually not a problem, because his farm is usually smaller than the resolution of weather data. That means that he can get all classification data for his farm in a day. But if he is from the colder regions of Slovenia he might still wait for some time before getting predictions.

In the future, the goal would be to add other weather features like precipitation or sun duration. Another important use case would be to focus on agriculturally more interesting plants like corn, wheat, and others. That would be important to ensure food security in years with the bad weather condition.

## 6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT program of the EC under project PerceptiveSentinel (H2020-EO-776115). The authors would like to thank Sinergise for their contribution to EO-learn library along with all the help with data analysis.

## References

- [1] Jan Dempewolf et al. “Wheat Yield Forecasting for Punjab Province from Vegetation Index Time Series and Historic Crop Statistics”. In: *Remote Sensing* 6 (Oct. 2014), pp. 9653–9675.
- [2] ECMWF. <https://www.ecmwf.int/>. Accessed 25 August 2019.
- [3] ESA. [https://www.esa.int/Our\\_Activities/Observing\\_the\\_Earth/Copernicus/Sentinel-2/Satellite\\_constellation](https://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-2/Satellite_constellation). Accessed 13 August 2018.
- [4] ESA. <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels/level-2>. Accessed 13 August 2018.
- [5] Cristina Gómez, Joanne C. White, and Michael A. Wulder. “Optical remotely sensed time series data for land cover classification: A review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 116 (2016), pp. 55–72. ISSN: 0924-2716.
- [6] Wuttichai Gunnula et al. “Normalized difference vegetation index relationships with rainfall patterns and yield in small plantings of rain-fed sugarcane”. In: *Australian Journal of Crop Science* 5 (Dec. 2011), pp. 1845–1851.
- [7] Toshichika Iizumi. “How do weather and climate influence cropping area and intensity?” In: *Global Food Security* (Nov. 2014).
- [8] University of Illinois at Urbana-Champaign. *Illinois agronomy handbook*. 24th ed. Urbana, Ill. : University of Illinois at Urbana-Champaign, College of Agricultural, Consumer and Environmental Sciences, Dept. of Crop Sciences, University of Illinois Extension, 1999. ISBN: 1883097223.
- [9] Filip Koprivec, Matej Čerin, and Klemen Kenda. “Crop classification using PerceptiveSentinel”. In: (Oct. 2018).
- [10] Ministrstvo za kmetijstvo gozdarstvo in prehrano. <http://rkg.gov.si/GERK/>. Accessed 25 August 2019.
- [11] Umer Saeed et al. “Forecasting wheat yield from weather data and MODIS NDVI using Random Forests for Punjab province, Pakistan”. In: *International Journal of Remote Sensing* 38 (Sept. 2017), pp. 4831–4854. DOI: 10.1080/01431161.2017.1323282.
- [12] Sinergise. <https://github.com/sentinel-hub/sentinelhub-py>. Accessed 14 August 2018.
- [13] Sinergise. <https://github.com/sentinel-hub/eo-learn>. Accessed 23 August 2019.
- [14] Silvia Valero et al. “Production of a Dynamic Cropland Mask by Processing Remote Sensing Image Series at High Temporal and Spatial Resolutions”. In: *Remote Sensing* 8(1) (2016), p. 55.
- [15] François Waldner, Guadalupe Sepulcre Canto, and Pierre Defourny. “Automated annual cropland mapping using knowledge-based temporal features”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 110 (2015), pp. 1–13. ISSN: 0924-2716.
- [16] Matteo Zampieri et al. “Wheat yield loss attributable to heat waves, drought and water excess at the global, national and subnational scales”. In: *Environmental Research Letters* 12 (June 2017), p. 064008.

# How overall coverage of class association rules affects the accuracy of the classifier?

Jamolbek Mattiev

Department of Information Sciences and Technologies  
University of Primorska  
Koper, Slovenia  
jamolbek.mattiev@famnit.upr.si

Branko Kavšek<sup>1,2</sup>

<sup>1</sup>Artificial Intelligence Laboratory  
Jožef Stefan Institute  
Ljubljana, Slovenia  
branko.kavsek@ijs.si

<sup>2</sup>Department of Information Sciences and Technologies  
University of Primorska  
Koper, Slovenia  
branko.kavsek@upr.si

## ABSTRACT

Associative classification (AC) is a data mining approach that combines classification and association rule mining to build classification models (classifiers). Experimental results show that in average the CBA-based approaches could achieve higher accuracy than some of the traditional classification methods.

In this paper, we focus on associative classification, where class association rules are generated and analyzed to build a simple, compact, understandable and relatively accurate classifier. Furthermore, we discuss how overall coverage and average rule coverage of such classifiers affect their classification accuracy. We compare our method that uses constrained exhaustive search with some “classical” classification rule learning algorithm that uses greedy heuristic search on accuracy in some “real-life” datasets. We have performed experiments on 11 datasets from UCI Machine Learning Database Repository.

Experimental evaluation shows that with decreasing overall coverage our proposed method tends to get slightly worse classification accuracy than the “classical” classification rule learning algorithms. Otherwise, the accuracy is similar or on some datasets even better than Naive Bayes and C4.5. On the other hand, the average rule coverage of our proposed method seems to have no effect on classification accuracy.

## CCS CONCEPTS

- Computing methodologies → Machine learning → Machine learning approaches → Rule learning
- Computing methodologies → Machine learning → Cross-validation
- Computing methodologies → Machine learning → Learning paradigms → Supervised learning → Supervised learning by classification

## KEYWORDS

Attribute, frequent Itemset, Minimum Support, Minimum Confidence, Class Association Rules (CAR), Associative Classification.

## 1 INTRODUCTION

Frequent patterns and their corresponding association rules characterize interesting relationships between attribute conditions and class labels, and thus have been recently used for effective classification. Association rules show strong associations between attribute-value pairs (or items) that occur frequently in a given dataset. Association rules are commonly used to analyze the purchasing patterns of customers in stores. Such analysis is useful in many decision-making processes, such as product placement, catalog design, and cross-marketing. The discovery of association rules is based on frequent itemset mining.

Associative classification mining is a promising approach in data mining that utilizes the association rule discovery techniques to construct classification systems, also known as associative classifiers. In the last few years, a number of associative classification algorithms have been proposed such as CBA: Classification based Association [11], CMAR: Classification based on Multiple Association Rules [10], CPAR: Classification based on Predicted Association Rule [13]. These algorithms employ several different methods, such as rule discovery, rule ranking, rule pruning, rule prediction and rule evaluation. Machine learning is one of the main phases in knowledge discovery from databases, which extracts useful patterns from data. Associative classification (AC) is lately among the focus areas in machine learning. AC integrates two known data mining tasks, association rule discovery and classification. The main aim is to build a model (classifier) for the purpose of prediction. Classification and association rule discovery are similar tasks in data mining, with the exception that the main aim of classification is the prediction of class labels, while association rule discovery describes correlations between items in a transactional database. In the last few years, association rule discovery methods have been successfully used to build accurate classifiers, which have resulted in a branch of AC mining. Several studies [4,9,10,11] have proved that AC algorithms are able to extract classifiers competitive with those produced by decision trees [3,12], rule induction [5,6,8] and probabilistic approaches [2].

In comparison with some traditional rule-based classification approaches, associative classification has two main characteristics. Firstly, it generates a large number of association classification rules. Secondly, support and confidence thresholds are applied to evaluate the significance of classification association rules. However, associative classification has some weaknesses. First, it often generates a very large number of

classification association rules in association rule mining, especially when the training dataset is large. It takes great efforts to select a set of high quality classification rules among them. Second, the accuracy of associative classification depends on the setting of the minimum support and the minimum confidence. Unbalanced datasets may heavily affect the accuracy of the classifiers. Third, the efficiency of associative classification may be also low, when the minimum support is set to be low and the training dataset is large. Although associative classification has some drawbacks, it can achieve higher accuracy than rule and tree-based classification algorithms on certain real life datasets. In this paper, we propose a simple classification method that selects a reasonable number of rules for classification, then, we find the overall coverage and average rule coverage of the classifier. We perform experiments on 11 datasets from the UCI Machine Learning Database Repository [7] and compare the results with some of the well-known classification algorithms (Naïve Bayes [2], PART [8], Ripper [6], C4.5 [12]).

## 2 PRELIMINARY CONCEPTS

Association rules consist of two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent. Association rules are generated from frequent itemset by analyzing the dataset and *support* and *confidence* thresholds are used to identify the most important relationships. *Support* is an indication of how frequently the items appear in the dataset. *Confidence* indicates the number of times the if/then statements have been found to be true.

Associative classification is a special case of association rule discovery in which only the class attribute is considered in the rule's right-hand side (consequent), for example, in a rule such as  $X \rightarrow Y$ ,  $Y$  must be a class attribute. One of the main advantages of using a classification based on association rules over classic classification approaches is that the output of an AC algorithm is represented in simple if-then rules, which makes it easy for the end-user to understand and interpret it

Let  $D$  be a dataset with  $n$  attributes  $\{A_1, A_2, \dots, A_n\}$  that are classified into  $M$  known classes and  $|D|$  objects. Let  $Y = \{y_1, y_2, \dots, y_m\}$  be a list of class labels. A specific value of an attribute  $A_i$  and class  $Y$  is denoted by lower-case letters  $a_{im}$  and  $y_j$  respectively.

**Definition 1.** An itemset is a set of some pairs of attributes and a specific value, denoted  $\{(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), \dots, (A_{im}, a_{im})\}$ .

**Definition 2.** A class association rule  $R$  has the form  $\{(A_{i1}, a_{i1}), \dots, (A_{im}, a_{im})\} \rightarrow y_j$  where  $\{(A_{i1}, a_{i1}), \dots, (A_{im}, a_{im})\}$  is an itemset and  $y_j \in Y$  is a class label.

**Definition 3.** The support count  $SuppCnt(R)$  of a rule  $R$  in  $D$  is the number of records of  $D$  that match  $R$ 's antecedent (left-hand side).

**Definition 4.** The support of rule  $R$ , denoted by  $Supp(R)$ , is the number of records of  $D$  that match  $R$ 's antecedent and are labeled with  $R$ 's class.

**Definition 5.** The confidence of rule  $R$ , denoted by  $Conf(R)$ , is defined as follows:  $Conf(R) = Supp(R)/SuppCnt(R)$ .

## 3 PROBLEM DEFINITION

Our proposed research assumes that the dataset is a normal relational table which has  $N$  examples described by  $L$  distinct attributes. These  $N$  examples are classified into  $M$  known classes. An attribute can be categorical (or nominal) or continuous (or numeric). In this paper, we treat all the attributes uniformly. Categorical attribute's values are mapped to a set of consecutive positive integers. Numeric attributes are discretized into intervals (bins), and the intervals are also mapped to consecutive positive integers. Discretization methods will not be discussed in this paper as there are many existing algorithms in the machine learning literature that can be used.

Our first goal is to generate the complete set of strong class association rules that satisfy the user-specified minimum support and minimum confidence constraints, and the second goal is to extract a reasonable number of strong CARs by pruning to build a simple and accurate classifier, the third and main goal is to find the overall coverage, average rule coverage and accuracy of the intended classifier.

## 4 OUR PROPOSED METHOD

Our proposed method consists of three steps. Firstly, a complete set of strong class association rules is generated from the given dataset. We then select a reasonable number of strong rules to build our simple and accurate classifier in the second step. Finally, we find the overall coverage, average rule coverage and accuracy of the classifier.

### 4.1 Generating class association rules

Association rule generation is usually split up into two separate steps:

1. First, we find all the frequent itemsets in a dataset by applying minimum support threshold. This step is the most important one, because, if minimum support is set to low, then we may have huge number of rules that lead to combinatorial complexity. If minimum support is set to high, then we may lose some interesting or strong rules, therefore, appropriate minimum support must be applied by analyzing the dataset.

2. Second, minimum confidence constraint is applied to generate strong class association rules from these frequent itemsets generated in the first step.

The second step is straightforward, that is why, we pay more attention to the first step. Apriori is a seminal algorithm described in [1] and it is mostly suggested for mining frequent itemsets.

Once the frequent itemsets from the dataset have been found, it is straightforward to generate strong class association rules from them (where strong CARs satisfy both minimum support and minimum confidence constraints). This can be done using following equation for confidence:

$$confidence(A \rightarrow B) = \frac{support\_count(A \cup B)}{support\_count(A)}. \quad (1)$$

The equation (1) is expressed in terms of itemsets support count, where  $A$  is premises (itemsets that is left-hand side of the rule),  $B$  is consequence (class label that is right-hand side of the rule),  $support\_count(A \cup B)$  is the number of transactions containing the itemsets  $A \cup B$ , and  $support\_count(A)$  is the number of

How overall coverage of class association rules affects the accuracy of the classifier?

transactions containing the itemsets  $A$ . Based on this equation, CARs can be generated as follows:

- For each frequent itemset in  $L$  and class label  $C$ , generate all nonempty subsets of  $L$ .
- For every nonempty subset  $S$  of  $L$ , output the rule “ $S \rightarrow C$ ” if  $\frac{\text{support\_count}(L)}{\text{support\_count}(S)} \geq \text{min\_conf}$ , where  $\text{min\_conf}$  is the minimum confidence threshold.

#### 4.2 Building our proposed classifier

We build our intended simple classifier by extracting the reasonable number of strong class association rules (already satisfied the minimum support and confidence requirements) that are generated in 4.1. Our proposed method is outlined in Algorithm 1.

---

##### Algorithm 1: Simple and accurate classification algorithm

---

**Input:** a set of CARs with their *support* and *confidence* constraints

**Output:** a subset of rules for classification

```

1:  D= Dataset();
2:  F= frequent_itemsets(D);
3:  R= genCARs(F);
4:  R= sort(R, minconf, minsup);
5:  G=Group(R);
6:  for (k=1; k≤ numClass; k++) do begin
7:      X= extract(class[k], numrules);
8:      Classifier= Classifier.add(X);
9:  end
10: for each rule y ∈ Classifier do begin
11:     if y classify new_example then
12:         class_count[y.class]++;
13:     end
14:     if max(class_count)=0 then
15:         predicted_class=majority_class(D);
16:     else predicted_class=index_of_max(class_count);
17:     return predicted_class

```

---

In lines 1-2 find all frequent itemsets in the dataset by using the Apriori algorithm. Line 3 generates the strong class association rules that satisfy the minimum support and confidence constraints from frequent itemsets. In line 4, CARs are sorted by *confidence* and *support* in descending order as follow:

Given two rules  $R_1$  and  $R_2$ ,  $R_1$  is said having higher rank than  $R_2$ , denoted as  $R_1 > R_2$ ,

- If and only if,  $\text{conf}(R_1) > \text{conf}(R_2)$ ; or
- If  $\text{conf}(R_1) = \text{conf}(R_2)$  but,  $\text{supp}(R_1) > \text{supp}(R_2)$ : or
- If  $\text{conf}(R_1) = \text{conf}(R_2)$  and  $\text{supp}(R_1) = \text{supp}(R_2)$ ,  $R_1$  has fewer attribute values in its left-hand side than  $R_2$  does;

Line 5 defines how to group the class association rules by their class labels (for example, if the class has three values, then, rules are grouped into three groups). In lines 6-9, we extract the reasonable number of rules per class that are equal to *numrules* to

SiKDD 2019, October 2019, Ljubljana, Slovenia

form a simple and accurate classifier. These set of rules become our final classifier. In lines 10-13, classification is performed by extracted CARs in line 6-9, if the rule can classify the example correctly, then, we increase the corresponding class count by one and store it. In lines 14-17, if none of the rules can classify the example correctly, then, algorithm returns the majority class value for the training dataset. Otherwise, it returns the majority class value of correctly classified rules.

#### 4.3 Overall coverage and average rule coverage

After our classifier is built in 4.2, it is straightforward to compute the overall coverage and average rule coverage of the classifier. To compute the overall coverage, we count the transactions that are covered by the classifier and divide it to total number of transactions in dataset. For the rule coverage, we count all the transactions that are covered by each rule in classifier and we take the average of them divided by total transactions.

---

##### Algorithm 2: Overall and average rule coverage of the classifier

---

**Input:** dataset and classifier

**Output:** overall coverage and average rule coverage

```

1:  n=D.length();
2:  C= Classifier;
3:  fill(classified_example)=false;
4:  for (i=1; i≤ C.length(); i++) do begin
5:      for (j=1; j≤ n; j++) do begin
6:          if C[i].premise classifies D[j].premise then
7:              rulecover[i]++;
8:              classified_example[j]=true;
9:      end
10:     avg_rulecover=avg_rulecover+ rulecover[i]/n;
11: end
12: for (i=1; i≤ n; i++) do begin
13:     if classified_example[i] then
14:         count++;
15: end
16: Overallcover_dataset=count/n;
17: return Overallcover_dataset, avg_rulecover

```

---

First line finds the length of the dataset. We form our classifier introduced in 4.2 (method is already created in lines 6-9 of algorithm 1) from the intended dataset in line 2. In the third line, we fill all initial values of *classified\_example* array as false. Lines 4-11 generally find the average rule coverage of the classifier. More precisely, we try to classify all the examples in the dataset by our classifier in lines 5-9. If rule's premise (left hand-side of the rule) classifies the example's premise (left hand-side of the example) in the dataset, then we increase the count for that rule's coverage and we mark that example as classified (this helps to compute the overall coverage of the dataset). Line 10 calculates the average rule coverage. We count all correctly classified examples in the dataset in lines 12-15 and overall coverage of the dataset is found in line 16. Line 17 returns the overall coverage and average rule coverage.

## 5 EXPERIMENTAL RESULTS

To find out the overall coverage, average rule coverage and to compare our results with some existing well-known classification methods on accuracy, we performed experiments on 11 real-life datasets from the UCI Machine Learning Database Repository. We used the WEKA software to explore the classification methods and 10 times random-split method (average result is

taken over 10 experiments) is used to perform experiments for both our method and other classification methods. In order to get enough rules for each class value and achieve a reasonable overall coverage, the parameter “#Rules per class” was set to 50 for all experiments. For other classification algorithms, Naive Bayes (NB), C4.5, PART (PT) and JRip (JR), we set up the default parameters.

**Table 1. Overall coverage and average rule coverage**

Dataset	# attr	# Cls	# rees	Min sup (%)	Min conf (%)	#Rules per class	Overall coverage (%)	Avg. rule coverage (%)	Accuracy (standard deviation) (%)				
									SA	C4.5	PT	JR	NB
Breast.Cancr	10	2	286	5	70	50	77.2	6.83	74.8(3.1)	72.0(3.5)	69.9(2.7)	68.9(4.4)	72.7(2.9)
Vote	17	2	435	1	80	50	93.1	28.86	95.4(2.4)	95.1(1.8)	95.5(1.4)	95.5(1.1)	89.1(1.9)
Balance.Sc	5	3	625	1	80	50	87.6	3.04	80.2(2.5)	67.2(2.4)	77.3(3.2)	77.4(2.0)	91.9(2.2)
Car.Evn	7	4	1728	0.8	70	50	76.2	7.14	81.4(2.8)	89.5(1.5)	95.0(1.5)	83.4(2.5)	84.8(0.9)
Tic-tac-toe	10	2	958	3	80	50	71.9	2.67	84.4(2.4)	84.7(3.2)	89.3(2.8)	97.5(0.6)	69.9(1.9)
Nursary	9	5	12960	2	60	50	98.0	3.78	88.6(2.6)	96.2(0.4)	98.7(0.4)	95.9(0.3)	90.4(0.4)
Hayes	6	3	160	1	50	50	100.0	5.56	80.1(7.1)	76.0(4.2)	73.3(7.7)	79.3(5.5)	79.7(7.9)
Mushroom	23	2	8124	20	80	50	84.4	4.76	68.2(1.6)	68.1(0.8)	64.3(0.7)	68.8(2.9)	69.7(0.5)
Lymp	19	4	148	3	70	50	81.0	18.76	75.3(6.4)	80.0(3.6)	79.0(6.9)	81.0(6.7)	85.1(4.1)
Monks	7	2	554	1	70	50	93.0	2.93	94.3(2.2)	98.4(2.7)	98.4(2.4)	98.4(2.2)	96.2(2.0)
Spect	23	2	267	0.5	60	50	81.4	27.21	78.6(3.1)	70.6(2.3)	67.1(5.3)	70.2(3.3)	69.9(4.1)
Average							85.8	10.14	81.9(3.3)	81.6(2.0)	82.5(3.2)	83.3(2.9)	81.8(2.6)

By analyzing the table of results (Table 1) we can observe that our classifier achieved better average accuracy than C4.5 and Naïve Bayes (81.9, 81.6 and 81.8 respectively). Standard deviations were higher for all methods on “Hayes” and “Lymp” datasets, that is, the differences between accuracies fluctuated and were reasonable high in 10 times random-split experiments. The overall coverages were lower than 80% on “Breast cancer”, “Car evaluation” and “Tic-tac-toe” and in those cases also the accuracy is slightly worse than that of the “classical” classifiers. On almost all other datasets our method achieves similar or slightly better accuracy. On the other hand, average rule coverage is surprisingly high on “Vote”, “Lymp” and “Spect” datasets, but seems to have no effect on classification accuracy.

## 6 CONCLUSION AND FUTURE WORK

Our comparison on selected 11 UCI ML datasets shows that with decreasing overall coverage our proposed method tends to get slightly worse classification accuracy than the “classical” classification rule learning algorithms. This fact is not surprising, since uncovered examples get classified by the majority classifier. When the overall coverage is above 85%, the accuracies of our classifier is similar or (on some datasets) even better than Naive Bayes and C4.5. On the other hand, the average rule coverage of our proposed method seems to have no effect on classification accuracy.

This research shows that overall rule coverage should be considered when selecting (pruning) the “appropriate” class association rules which we plan to implement in future research.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program

and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund).

## REFERENCES

- [1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487-499. Chile (1994).
- [2] Baralis, E., Cagliero, L., Garza, P.: A novel pattern-based Bayesian classifier. IEEE Transactions on Knowledge and Data Engineering 25(12), 2780-2795 (2013).
- [3] Breiman, L.: Random Forests. Machine Learning 45(1), pp. 5-32 (2001).
- [4] Chen, G., Liu, H., Yu, L., Wei, Q., Zhang, X.: A new approach to classification based on association rule mining. Decision Support Systems 42(2), 674-689 (2006).
- [5] Clark, P., Niblett, T.: The CN2 induction algorithm. Machine Learning, 3(4), 261-283 (1989).
- [6] Cohen, W., W.: Fast Effective Rule Induction. In: ICML'95 Proceedings of the Twelfth International Conference on Machine Learning, pp. 115-123, Tahoe City, California (1995).
- [7] Dua, D., Graff, C.: UCI Machine Learning Repository, Irvine, CA: University of California (2019).
- [8] Frank, E., Witten, I.: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, pp. 144-151. USA (1998).
- [9] Hu, L., Y., Hu, Y., Han, T., Tsai, C. F., Wang, J. S., Huang, M. W.: Building an associative classifier with multiple minimum supports, SpringerPlus, 5:528, (2016).
- [10] Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. in Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01), pp. 369-376, San Jose, California, USA (2001).
- [11] Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. in Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD '98), pp. 80-86, New York, USA (1998).
- [12] Quinlan, J.: C4.5: Programs for Machine Learning, Machine Learning 16(3), 235-240 (1993).
- [13] Xiaoxin, Y., Jiawei, H. CPAR: Classification based on Predictive Association Rules. Proceedings of the SIAM International Conference on Data Mining, pp. 331-335, San Francisco, U.S.A (2003).



# Epileptic Seizure Detection Using Topographic Maps and Deep Machine Learning

Patrik Kojanec  
Department of Information  
Sciences and Technologies  
University of Primorska  
Koper, Slovenia  
patrik.kojanec@gmail.com

Branko Kavšek  
Department of Information  
Sciences and Technologies  
University of Primorska  
Koper, Slovenia  
branko.kavsek@upr.si

César A. D. Teixeira  
Centre for Informatics and  
Systems  
University of Coimbra  
Coimbra, Portugal  
cteixe@dei.uc.pt

## ABSTRACT

One third of all epileptic patients is resistant to medical treatment. The construction of machines, that would detect an imminent epileptic attack based on EEG signals, represents an efficient alternative, that would help to increase their quality of life. In this article we described the implementation of an automatic detection method, based on the signal of different frequency sub-bands, using topographic maps and deep learning techniques. We constructed an ensemble of five convolutional neural networks, to classify samples of each sub-band and chose the final decision by a majority voting. The ensemble obtained 99.20% accuracy, 96.48% sensitivity and 99.27% specificity when detecting seizures of one patient. Moreover, when the networks were trained with samples taken randomly from the inter-ictal intervals, we identified on 18 of 21 seizures some false positive classifications close to the seizure onset, thus anticipating the detection of the seizure. Such misclassifications did not occur when training was performed with samples taken within five minutes of the seizure onset.

## KEYWORDS

Epileptic Seizure Detection, Deep Learning, Topographic Maps, Electroencephalogram.

## 1 Introduction

Epilepsy is a neurological disorder characterized by sudden seizure attacks, that may cause in patients loss of consciousness and motor control. It is esteemed that epilepsy affects about 50 million people world-wide and represents up to 1% of the global burden of disease[6]. Although in the last decades many anti-epileptic drugs (AEDs) have been introduced[4], to more than 30% of the patients these treatments are ineffective. Therefore, their daily life activities are very restricted because of the unpredictability of the attacks. The development of different approaches, that could timely inform patients of an imminent epileptic attack is necessary to increase their quality of life.

The most used tool to monitor brain's electrical activity is the electroencephalogram (EEG). However, due to the complexity of the EEG signals, visual detection of epileptic seizures from the signal often results misinterpreted or mistaken. Therefore, in the last decades much research has been oriented towards finding automated detection procedures, that would efficiently analyze

large chunks of signals, timely give out warnings and help the medical staff to deliver treatment on time[8].

Since the first studies of epilepsy seizures with EEG, it is known that an epileptic attack has a detectable electrical discharge in the brain (EEG onset), prior to the manifestation of convulsions, loss of consciousness and others symptoms (clinical onset)[7]. The time window between these events usually ranges between 0 to 30 seconds, sometimes reaching over 1 minute. Therefore, being able to detect early enough the EEG onset of the seizure could give enough time to the patient to get the treatment or at least to reach a safe environment.

Based on these motivations, we constructed the following model for epilepsy seizure detection, based on topographic maps generated from EEG signals and deep machine learning classifying techniques.

## 2 Experimental Setting

In this work we used data from the EPILEPSIAE database[5]. We selected a single patient, with a defined focal epilepsy in the temporal lobe. The recording of the patient of about 161.1 hours contained 22 seizures, averaging 3.28 seizures per day. However, one seizure was discarded from the study since it was described as not reliable. The sampling frequency of the machine was 256 Hz.

The work is composed by two studies, which follow the general processing pipeline: raw data is preprocessed and transformed from time to frequency domain, then the relative powers calculated from the signals of the frequency sub-bands are used to generate topographic maps, which are then fed to the classifier. After a regularization procedure, the performance of the model is evaluated. Figure 1 schematically describes the mentioned pipeline.

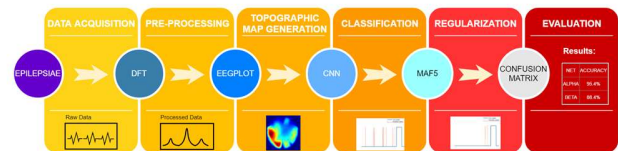


Figure 1: Processing pipeline of the DM process

### 2.1 Study 1: 80% overlap

#### 2.1.1 Pre-Processing and Feature Extraction

Raw data needs to undergo few pre-processing steps, before it can be used to generate topographic maps. Here, we used functions

from the EPILAB® [3] package. Next, with a high-pass filter kept the frequencies between 0.1Hz and the Nyquist frequency, which in this case was 128 Hz, and then a 50 Hz notch filter was applied to remove possible power line artifacts.

EEG signals are non-stationary. Many mathematical tools for analysis assume the stationarity of the signal. One way to enforce an “artificial” stationarity is by segmenting the signal and making the analysis of the segments globally valid[11]. Initially, the chosen length of segments (sliding time windows) was 5 seconds, with an overlap of 80%. This allows to assume stationarity in these five second windows and preserve frequency resolution. Furthermore, by overlapping by 80% we obtain four time more samples than we would have obtained without it and we might detect additional information that could not be captured between the end of a window and the beginning of another. Five seconds window's length represent a good compromise to keep sufficient time and frequency resolutions and is often used in EEG analysis [2].

While pre-processing the data in time windows, 5 basic frequency features were extracted. Features corresponded to the relative powers of different frequency sub-bands obtained with the Discrete Fourier Transform (delta, theta, alpha, beta, gamma).

### 2.1.2 Topographic Map Generation

Topographic maps were generated using the *eegplot* function by I. Silva [9], publicly available on MATLAB Exchange. A map was generated for every seizure timepoint, for all five features, resulting in 929 samples for each feature (dataset A). To balance the datasets, the same number of non-seizure samples was generated with randomly selected timepoints. Both sets of samples were further on divided for training (80%), validation during training (10%) and testing (10%).

A second testing set was generated (dataset B), with non-seizure samples taken every second in series from the five-minute interval prior the seizure and half the number of seizure samples after the seizure. Unfortunately, due to limited data, the seizure samples were the same as the one used for training. The new testing set had 6775 non-seizure samples and 929 seizure samples. A clear representation of the datasets is shown in Table 1.

**Table 1: Representation of the datasets**

Dataset	Objective	Description
A	Training	- 929 ictal samples
	+ Testing	- 929 non-ictal samples taken randomly Separation of data: 80% train., 10% valid., 10% test
B	Testing	- 929 ictal samples (same as data set A) - 6775 non-ictal samples (5 min before the seizure + half the number of seizure samples, after the seizure)

### 2.1.3 Training and Classification

The classifier we used was an ensemble of five convolutional neural networks, one for each feature. All networks had the same topography, however they differed in the hyperparameters' value. The best hyperparameters for the networks were selected after a

grid search on the initial training set. An example of the network structure is shown in Table 2.

**Table 2: Example of a network structure**

Layer:	Name	Output	Learnables
1	InputLayer	766x884x3	0
2	Conv1	383x442x16	448
3	BatchNorm1	383x442x16	32
4	ReLu1	383x442x16	0
5	MaxPool1	383x442x8	0
6	Conv2	383x442x8	520
7	BatchNorm2	192x221x4	8
8	ReLu2	383x442x8	0
9	DropOut1	383x442x8	0
10	MaxPool2	192x221x8	0
11	Conv3	192x221x4	132
12	BatchNorm3	192x221x4	8
13	ReLu3	192x221x4	0
14	DropOut2	192x221x4	0
15	FullCon1	1x1x32	5431328
16	ReLu4	1x1x32	0
17	FullCon2	1x1x2	66
18	SoftMax	1x1x2	0
19	ClassOutput		0

The networks were trained for 32 epochs, using the RMSprop optimizer, randomly shuffled minibatches of 16 samples and the training performance was validated every 30 iterations. The same networks were also used to test the second testing set.

## 2.2 Study 2: 98% overlap

### 2.2.1 Pre-Processing and Feature Extraction

Due to the limited ictal data in the first study, we decided to perform a second one with more samples. To augment the data, we increased the overlap to 98%, which produced ten times more samples. Besides the overlap, all the pre-processing steps were performed identically as in the first study.

### 2.2.2 Topographic Map Generation

In the second study the training samples were not selected randomly as in the first study, but they were picked from the intervals from five to one minute prior every seizure (dataset C). We intentionally kept the last minute out of training, with the intent of obtaining again the FP classifications close to the seizure onset. Furthermore, to balance the seizure and non-seizure datasets we added some more randomly picked non-seizure samples.

Next, similarly as in the first study, we generated another testing set with the samples taken in series, starting from one hour before the seizure (dataset D). However, in the second study the samples were taken every two seconds, due to the notorious computational overhead. Again, a better representation of the datasets is shown in Table 3.

**Table 3: Description of used datasets**

Dataset	Objective	Description
C	Training	- 8342 ictal samples - 5280 non-ictal samples (5-1 min before seizure) - 3062 non-ictal samples (taken randomly) 20% of the samples were used for validation
D	Testing	- 950 ictal samples - 37800 non-ictal samples (1 hour before seizure, samples taken every two seconds)

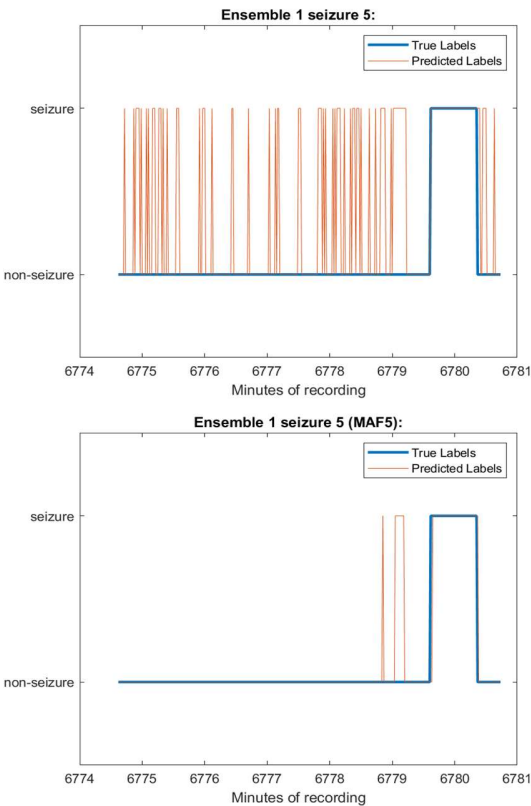
### 2.2.3 Training and Classification

In the second study we used the same topography of the classifier as in the first study, however the networks were trained with the new training set (dataset C). We opted to use the same hyperparameters as in the first study, since we used the same sub-bands and the same seizures of the same patient. We used the same training procedure, apart from the number of epochs and validation frequency, which were set to 16 and 500 respectively, due to the augmented data.

## 3 Results

### 3.1 Study 1: 80% overlap

The ensemble increased almost all the classification scores, comparing single individual networks, apart from the sensitivity in



**Figure 2: The effect of applying the MAF5 filter**

the theta sub-band network, which was originally higher than in the ensemble. The evaluation metrics are shown in Table 4.

**Table 4: Evaluation metrics**

	TP	TN	FP	FN	AC	SS	SP
Alpha	85	77	16	8	87.10%	91.40%	82.80%
Beta	86	79	14	7	88.71%	92.47%	84.95%
Gamma	80	86	7	13	89.25%	86.02%	92.47%
Delta	80	80	13	13	86.02%	86.02%	86.02%
Theta	89	88	5	4	95.16%	<b>95.70%</b>	94.62%
Ensemble 1	88	91	2	5	<b>96.24%</b>	94.62%	<b>97.85%</b>

When tested on the second set of samples, the specificity dropped to 89.88%, meaning that on this interval there was an increase of FPs. Since the samples were selected in series, we decided to apply a moving average filter (MAF5) to reduce FP predictions. Both AC and SP increased, while the SS dropped. These results are presented in Table 5. Furthermore, in Figure 2 is presented the effect of the MAF5 filter.

After applying the MAF5 filter, on 18 of 21 seizures we could identify FP classifications, within 1 minute before the seizure (see Figure 2). Such misclassifications are promising, since they suggest the model could even anticipate the seizure onset. This was also a reason, that led us perform a second test.

**Table 5: Results of first study**

	AC	SS	SP
Normal	90.99%	99.65%	89.88%
<b>MAF5</b>	<b>96.68%</b>	<b>92.56%</b>	<b>97.23%</b>

### 3.2 Study 2: 98% overlap

In the second study we tested the networks directly on the dataset D. After applying the MAF5 filter, the ensemble obtained 99.20% accuracy, 96.48% sensitivity and 99.27% specificity, as shown in Table 6.

**Table 6: Results of second study**

	AC	SS	SP
Normal	85.34%	98.96%	84.99%
<b>MAF5</b>	<b>99.20%</b>	<b>96.48%</b>	<b>99.27%</b>

We also noticed that the FP classifications close to the seizure onset did not occur in the second study. Moreover, they appeared further away from the seizure, mostly from 40 minutes to 10 minutes before (see ).

## 4 Discussion

The first ensemble increased almost all the classification scores compared to individual networks. However, when tested with a series of samples taken close from the seizure onset, the number of false positives increased, while the seizures remained correctly classified. We believe that this high classification score of the

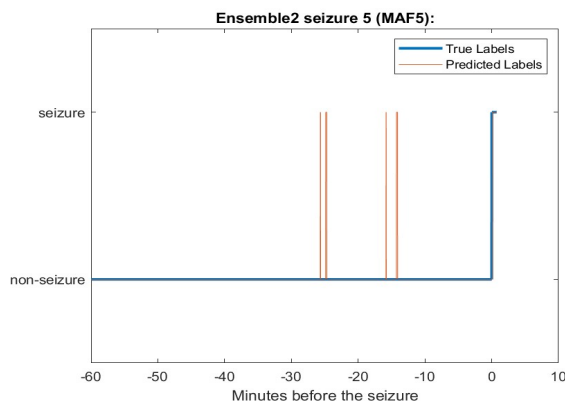


Figure 3: FPs appearing far from the seizure onset

seizure set was due the fact that 80% of the seizure samples used for this testing, were also the same used for training the network. Therefore, it is believed, that the network “remembered” those samples and classified them correctly.

The high increase of false positive classifications with the new test set are believed to be due the fact that the samples used for training, which are taken randomly from all the non-ictal intervals, were distinct from the ones really close the seizure onset. This leads to believe, that there is a noticeable change in the signal while approaching the transition from the non-ictal to the ictal phase. However, by applying the MAF5 filter, the score increased. The MAF5 filter could potentially suit the real-time detection, since it only requires a delay of few samples. In the case of 5 second sliding windows, with 80% overlap, this delay would be of 2 seconds with the filter size 5, which fits early detection necessities.

After applying a moving average filter, with a stride of five, to reduce the number of false positive classifications, we noticed that for 18 out of all 22 seizures, within a range of 1 minute, some false positive classifications persisted (as seen in Figure 2). This is curious, since these false predictions could anticipate the occurrence of an imminent attack. Thus, we retrained the networks with augmented data, to see if these false predictions persist.

After testing the second ensemble on the dataset D, we could not identify the false alarms close to the seizure onset as in the previous testing. Moreover, only four seizures had a false alarm within five minutes before the seizure. We believe that this is a consequence of the training set C. Although samples from one minute before the onset were left out of training, with the intention of producing some false alarms, they were classified correctly in the test. Another evidence supporting our claim are the false predictions far from the seizure, that appear in most of the seizures. Hence, to get rid of them and produce the false predictions close to the seizure onset, the training set should include more samples that are far away from the seizure. However, these results show that there is a difference within non-ictal samples far from the seizures and non-ictal samples close to them.

## 5 Conclusions

As expected, the second study’s classification scores outperformed the first one, since it was trained on a larger dataset.

It obtained a 99.20% accuracy, 96.48% sensitivity and 99.27% specificity. However, it failed to replicate the FP predictions close to the seizure onset, as the first one did.

Overall, the model obtained scores that are comparable to the state-of-the-art results[1,8,10]. Although this model does not have an early prediction performance, it still yields good detection scores. Furthermore, both studies give some insights on the early detection, that might be possible to perform, due to the diversity of the non-ictal samples, located far and close to the seizure.

## ACKNOWLEDGMENTS

The first author would like to thank his coordinators, University of Coimbra, University of Primorska and the CISUC Clinical Informatics laboratory for the support throughout his studies and research.

The second author gratefully acknowledges the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European regional Development Fund).

## REFERENCES

- [1] Saleh A Alshebeili, Fathi E Abd El-Samie, Tariq Alshawi, Turkey N Alotaiby, and Ishtiaq Ahmad. 2015. EEG seizure detection and prediction algorithms a survey. *EURASIP J. Adv. Signal Process.* 2014, 1 (2015). DOI:https://doi.org/10.1186/1687-6180-2014-183
- [2] Bruno Direito, César Teixeira, Bernardete Ribeiro, Miguel Castelo-branco, and Francisco Sales. 2012. Modeling epileptic brain states using EEG spectral analysis and topographic mapping. *J. Neurosci. Methods* 210, 2 (2012), 220–229. DOI:https://doi.org/10.1016/j.jneumeth.2012.07.006
- [3] A. Dourado, R.P. Costa, B. Schelter, S. Nikolopoulos, C.A. Teixeira, M. Le Van Quyen, C. Alvarado-Rojas, B. Direito, H. Feldwisch-Drentrup, M. Valderrama, and J. Timmer. 2011. EPILAB: A software package for studies on the prediction of epileptic seizures. *J. Neurosci. Methods* 200, 2 (2011), 257–271. DOI:https://doi.org/10.1016/j.jneumeth.2011.07.002
- [4] Shery Jacob and Anroop B Nair. 2016. An Updated Overview on Therapeutic Drug Monitoring of Recent Antiepileptic Drugs. *Drugs R. D.* 16, 4 (December 2016), 303–316. DOI:https://doi.org/10.1007/s40268-016-0148-6
- [5] Juliane Klatt, Hinnerk Feldwisch-drentrup, Matthias Ihle, Vincent Navarro, Markus Neufang, Cesar Teixeira, Claude Adam, Mario Valderrama, Adrien Witon, Michel Le Van Quyen, Francisco Sales, Antonio Dourado, Jens Timmer, Andreas Schulze-bonhage, and Bjoern Schelter. 2012. The EPILEPSIAE database: An extensive electroencephalography database of epilepsy patients. 53, 9 (2012), 1669–1676. DOI:https://doi.org/10.1111/j.1528-1167.2012.03564.x
- [6] World Health Organization. 1970. *Atlas epilepsy care in the world*. World Health Organization. Retrieved from http://www.who.int/training/1066543298
- [7] W Penfield and T C Erickson. 1941. *Epilepsy and cerebral localization*. Charles C. Thomas, Oxford, England.
- [8] Jagriti Saini and Maitreyee Dutta. 2017. An extensive review on development of EEG-based computer-aided diagnosis systems for epilepsy detection. *Netw. Comput. Neural Syst.* 28, 1 (2017), 1–27. DOI:https://doi.org/10.1080/0954898X.2017.1325527
- [9] Ikaro Silva. 2013. *ceegplot*(.).
- [10] Ihsan Ullah, Muhammad Hussain, Emad-ul-haq Qazi, and Hatim Aboalsamh. 2018. An Automated System for Epilepsy Detection using EEG Brain Signals based on Deep Learning Approach Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland Visual Computing Lab, Department of Computer Science, College of Com. *Expert Syst. Appl.* 107 (2018), 61–71. DOI:https://doi.org/10.1016/j.jss.2016.08.088
- [11] Andrea Varsavsky, Iven Mareels, and Mark Cook. 2011. *Epileptic Seizures and the EEG*.

# Demand Forecasting for Industry 4.0: predicting discrete demand from multiple sources for B2B domain

Jože Martin Rožanec<sup>†</sup>

Qlector d.o.o.

Jožef Stefan Institute International  
Postgraduate School  
Ljubljana, Slovenia  
joze.rozanec@qlector.com

Dunja Mladenec

Jožef Stefan Institute

Jožef Stefan Institute International  
Postgraduate School  
Ljubljana, Slovenia  
dunja.mladenec@ijs.si

Blaž Fortuna

Qlector d.o.o.

Jožef Stefan Institute International  
Postgraduate School  
Ljubljana, Slovenia  
blaz.fortuna@qlector.com

## ABSTRACT

Demand is the amount of certain product required by buyers at a point in time. Demand forecasting tries to predict future demand based on available information. It is considered a key component of each manufacturing company since improvements on it translate directly to resources planning, stocks and overall operations.

In the context of Industry 4.0, industry digitalization provides an ever-increasing number of data sources which can be consumed to gain visibility over all operations and used to optimize different processes within it. This also opens new possibilities into the field of demand forecasting, where multiple data sources can be integrated to get timely data for accurate forecasts.

We describe an efficient approach for demand forecasting for discrete components B2B industry. The proposed approach provides as good or better forecasts as logisticians for most months in six months period and achieves savings considering all test months period.

## CCS CONCEPTS

- Information systems → Information systems applications → Enterprise information systems → Enterprise resource planning • Computing methodologies → Machine learning → Machine learning approaches
- Computing methodologies → Artificial intelligence

## KEYWORDS

demand forecasting, industry 4.0, B2B manufacturing, time series analysis

### ACM Reference format:

Jože Martin Rožanec, Dunja Mladenec and Blaž Fortuna. 2019. Demand Forecasting for Industry 4.0: predicting discrete demand from multiple sources for B2B domain. In *Proceedings of SiKDD (SiKDD'19)*. IS, Ljubljana, Slovenia.

## 1 INTRODUCTION

Demand forecasting is the task of predicting the number of units of a specific good for a given point of time in the future before we actually get all orders from the customers. It is a critical factor in just-in-time supply chains, where companies are expected to offer short lead times for products with complex production processes made of raw materials or components with longer lead times. In this paper we focus on solving this task by using machine learning techniques.

As a socioeconomic phenomenon there are many aspects that may enhance predictions when captured into features, such as economic context (does demand increase with economic growth, how it is affected by price changes, are there substitute products, what kind of market do we operate on), other context facts (marketing campaigns, fashionable features, product established in market or a new release) or inherent product properties

(product category, whether is perishable, etc.). By considering a wider context, we may mitigate demand signal distortions that happen at each new intermediary level of a supply chain, in what is known as the bullwhip effect [1]. Another factor of uncertainty is the forecasting horizon: further the horizon, less likely is to be the future similar to past and present state of matters and more difficult to be predicted accurately [2].

When considering forecasting techniques, it may be important to consider characteristics of demand. Authors discriminate demand along two main variables: by considering variability in demand timing and quantity. A classification scheme is described in Figure 1.

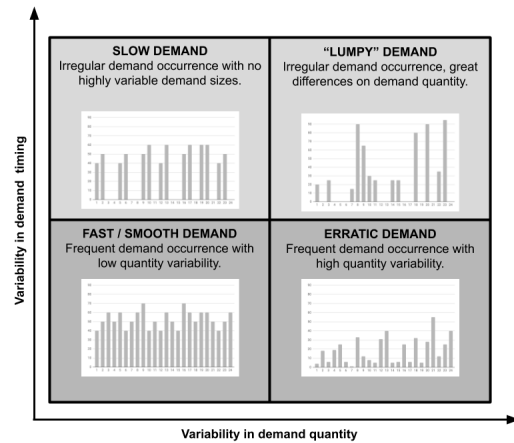


Figure 1: demand classification as per Williams et al. [3] and further elaborated in [4]

In our case we focus on demand forecasting for items from the B2B discrete manufacturing industry which are established in the market and sold under perfect market conditions. Since most of the products correspond to fast moving inventory, we do not discriminate between different demand types and treat all the products in the same way.

Publications addressing demand forecasting explored auto-regressive moving averages [5, 6, 7, 8], multiple linear regression [9] (MLR), Bayesian approaches [10], support vector regressors (SVR) [11] and artificial neural networks (ANN) [12]. In our research we consider naïve forecasting (last observed value as prediction), auto-regressive moving average (ARIMA), MLR, SVR and gradient boosted regression trees (GBRT) [13] models and compare them to logisticians predictions issued in two points in time: six weeks and three days before the event. We do not consider ANNs due to our limited amount of available data to train them.

The remainder of this paper is structured as follows. In Section 2, we define the problem, features, metrics and briefly describe forecasting techniques. Section 3 describe our dataset and preprocessing steps. In Section 4 we describe the experiments we conducted and results we obtained. Section 5 presents conclusions and directions for further work.



## 2 PROBLEM DEFINITION

Demand forecasting requires to predict the number of units for a product that will be ordered at a given future point in time. We consider different time horizons:  $H_1, \dots, H_n$ , and train a specific model for each of them. The goal is to make accurate predictions about future demand based on historical demand data, annual sales plans, open sales orders and some contextual information.

### 2.1 Features

The main features we obtain from datasets correspond to historical data describing observed demand for each product, high-level estimations such as annual forecasts (describe expected product demand over the year), low-level demand proxies such as open sales orders for a given point in time, and contextual data (economic indicators, prices for relevant raw materials, vacation periods at buyers and manufacturer companies).

Derivative features are meant to explore the relation between the original variables as well as how do they relate to each other in different points in time. This way they reflect the direction and magnitude of trends in comparison to previous months. Months immediately before the target date provide information about recent demand and context behavior, while values from the same months but considered a year before help to learn seasonality patterns where it may exist.

The annual sales forecast and open sales orders give us some insight to the expected future. The annual forecast displays total amount to be sold over the year and a projected sales distribution. Open sales orders give us a weak signal about expected demand and may help to better estimate the target value given the rest of the feature's context. Both can also be related to learn if projected sales accurately reflect the annual forecast, differ by some factor or may not follow original expectations at all. In a similar way we learn past relations between projected and real demand as well as the relation between open sales at a given point in time and later demand realization.

Since we have two forecasting horizons with a six weeks separation and data available at a monthly frequency, we are able to compute additional features for models aimed to predict three days before the event horizon.

### 2.2 Metrics

To measure forecast performance across models we chose the mean absolute error metric. This metric is not sensitive to occasional large errors, which is important in the context of demand forecasting, where at specific points in time demand may display abnormal behavior that cannot be forecasted. The model should not be strongly penalized on them when trained. The metric also provides a straightforward interpretation (errors are measured in the same units as data and error magnitudes directly correlate on how well/bad the model performs). This does not turn into an issue when comparing different models, since by working on same dataset, we measure all models in same units and magnitudes.

We use the same metric as objective and evaluation metric for models we train.

### 2.3 Prediction techniques

We take into account five types of forecasting techniques: naïve forecasting, autoregressive integrated moving average (ARIMA), multiple linear regression (MLR), support vector regressor (SVR) and gradient boosted regression trees (GBRT). ARIMA and MLR are widely used in the literature to forecast fast moving products, while gradient boosted regression trees, to the extent of our knowledge, were not applied to demand forecasting in the B2B manufacturing industry.

Naïve forecasting method considers that the value to take place at time  $t+1$  will be close to the one present at time  $t$  and thus the best proxy is to use the same value of time  $t$  as prediction. In our case we consider the last demand value we are able to observe given a time horizon as the output value of our prediction.

ARIMA is a stochastic time series method that grounds its predictions on three components: auto-regression (estimates white noise affecting the data by regressing the variable on own past values), integration (reduction of

seasonality and trend by differencing the time series) and moving average (considers previous values to estimate the target value).

Both, the naïve forecasting and ARIMA are limited only to demand forecasting historic values and cannot consider a broader context in their predictions.

MLR is a simple method that explains linear relationships between a continuous dependent variable and multiple independent ones. The independent variables may be continuous or one-hot encoded categorical ones.

SVR is a regression method based on support vectors, where a kernel is used to map low dimensional data into a higher dimension and then best hyperplane and boundary lines are computed to predict target values. The method allows to fit the error within a certain threshold. In our case we use a radial basis function kernel (RBF kernel), which helps us to consider non-linear relationships between features.

GBRT makes use of gradient boosting, which generalizes boosting to an arbitrary loss function, and uses regression trees to approximate the negative gradient. These are built iteratively, each tree representing a step of gradient descent when optimizing the loss function.

## 3 DATA DESCRIPTION

### 3.1 Dataset

Our dataset was provided by manufacturing B2B industries and contains information about 69 products over a period of 68 months.

Among features we have historic demand data for all products, annual demand plans and open sales orders when the forecast is issued. Our prediction target is the amount of a certain product to be demanded by buyers for a given month - on two prediction horizons: six weeks and three days ahead.

### 3.2 Data preparation

Given the original dataset, we first analyzed data density. We found that there are multiple products with scarce demand datapoints due to irregular demand or by the fact that started being produced later in time. Since demand points density may affect model results, we decided to create multiple datasets based on how many points of historical demand data do we have - all with identical features. This way for all experiments performed, we have datasets with 0+, 10+, 20+, 30+, 40+ and 50+ demand history points.

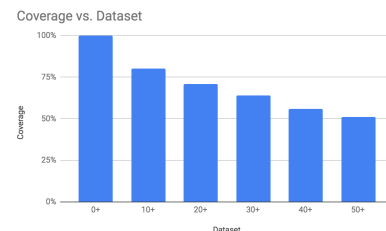


Figure 2: product coverage by dataset.

We then analyzed data distributions and observed that most features display Normal distributions when considering a single product, but over the whole dataset the distribution is lognormal. To mitigate this issue and differences in orders of magnitude, we first transformed them using a Yeo-Johnson transformation followed by standard scaling and a Min-Max transformation. The Yeo-Johnson transformation [14] ensures transformed values follow a Gaussian distribution, while the standard scaling centers them around zero with a standard deviation of one. By using the Min-Max transformation we get them into [0-1] range regardless of their original magnitudes. We also observed that some materials exhibit seasonality and trend but did not perform any ad-hoc preprocessing for them.

Among computed features, there are many that refer to past performance (same month or months close to it, for current year as well as the year

before). This cannot be computed where we lack enough history and thus decided to compute them and then prune the dataset to last 56 months to discard spurious values.

Considering that demand forecasting models are time sensitive, we use last six months for testing and devote the rest to train the models. We do so in such a way that the train set is not fixed, but we use all data up to the month to be predicted for training. By doing so, we had more records available to train models targeted towards last months and could ensure time proximity towards them.

We devote a month close to the test set as validation set. We performed an experiment to understand if excluding validation set the data from the train set affects results by degrading predictions or if including it causes the model to overfit. Results showed that including the validation set into the training set improved results without risk of overfitting and thus used this setup for the experiments.

The dataset with all features described is used for the MLR and GBRT models, while the naïve and ARIMA models use only historic demand data for a given product up to the month when the prediction shall be made.

## 4 EXPERIMENTS AND RESULTS

All experiments above were performed on datasets with demand records density of 0+, 10+, 20+, 30+, 40+ and 50+ records, to understand the tradeoff between data completeness and a greater number of records reflects in forecast results. In all cases we devote last six months to testing, and the rest of the data to train the model.

We use the following notation to describe models: *ModelName-FeatureSet-Transform-DatasetFiltering*

Valid *ModelName* values are SVR, MLR and GBRT; *FeatureSets* can be 3m, 6m, 9m and 12m – notating that features were computed over a window of three, six, nine or twelve months. *Transform* can be “wTT” if transforms were applied to dataset target, otherwise we use “nTT”. *DatasetFiltering* accepts three possible values: “2Y”, “3Y” or “4Y” indicating that the dataset contains train records for two, three or four years respectively plus six months of test data.

Results are expressed in error ratio, computed as:

$$\text{Model Error Ratio} = 1 - \frac{\text{MAE Model}}{\text{MAE logistician}}$$

### 4.1 Feature set comparison

First experiment we performed was to understand how many months we should consider when looking back to create features in order to make better predictions. To this purpose, we developed four sets of features, created with a time window of three, six, nine and twelve months from target date. When considering a six weeks horizon, we found out that best results were achieved by GBRT-9m-nTT-4Y and GBRT-6m-nTT-4Y, followed by GBRT-3m-nTT-4Y which accounts for half of second-best predictions. For a three-day horizon, most best results were achieved by GBRT-3m-nTT-4Y, making best prediction for half of datasets and second-best prediction for two of three remaining ones.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	9m	4Y	NO	0.11
10+	6m	4Y	NO	0.15
20+	12m	4Y	NO	0.11
30+	9m	4Y	NO	0.14
40+	6m	4Y	NO	0.14
50+	6m	4Y	NO	0.17

**Table 1:** best results when considering six weeks forecasting horizon. All of them run with GBRT algorithm.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	3m	4Y	NO	0.12
10+	3m	4Y	NO	0.17
20+	6m	4Y	NO	0.09
30+	3m	4Y	NO	0.15
40+	12m	4Y	NO	0.19
50+	9m	4Y	NO	0.17

**Table 2:** best results when considering three days forecasting horizon. All of them run with GBRT algorithm.

### 4.2 Target normalization

We then compared trained GBRT models against new ones where same transformations as applied to features were applied to target values. Our assumption was that by transforming the target, which had a lognormal distribution, we should get a better spread of predictions and better results. Most best results for six-weeks horizon were found at GBRT-6m-wTT-4Y and GBRT-9m-wTT-4Y models except for 10+ and 50+ datasets. When comparing models with and without target transform, most best results at models without target transform resulted in second best results if considered globally.

On the other hand, for three-day forecasting horizons, applying transformations to the target improved results most cases, but still half of best predictions could be found among models that do not require target transformation. In this context, GBRT-12m-wTT-4Y displayed best global performance for half of datasets considered.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	9m	4Y	YES	0.16
10+	12m	4Y	YES	0.18
20+	6m	4Y	YES	0.12
30+	6m	4Y	YES	0.15
40+	9m	4Y	YES	0.15
50+	9m	4Y	NO	0.17

**Table 3:** best results when considering six weeks forecasting horizon. All of them run with GBRT algorithm.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	12m	4Y	YES	0.15
10+	3m	4Y	NO	0.17
20+	12m	4Y	YES	0.11
30+	12m	4Y	YES	0.19
40+	12m	4Y	NO	0.19
50+	9m	4Y	NO	0.17

**Table 4:** best results when considering three days forecasting horizon. All of them run with GBRT algorithm.

### 4.3 Records history contribution

Since forecasting models are time sensitive, we explored if recent history is more relevant in such a way that older records may deteriorate forecasting results. We pruned the dataset removing all records older than two or three years in train set and compared models trained on them with those obtained from training on full history.

When analyzing a six-weeks horizon, we found out that pruning history leads to better results achieving almost all first- and second-best results globally. Best results were achieved by models with three years of history with best performance for GBRT-9m-wTT-3Y.

For a three-days horizon, we observed that GBRT models with different feature sets over pruned datasets performed worse than existing ones. Overall, we observe GBRT algorithm achieved best results with target transforms enhancing results on half datasets and that 12m was the most frequent feature set among competitive models.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	9m	4Y	YES	0.16
10+	12m	3Y	YES	0.22
20+	12m	2Y	YES	0.17
30+	9m	3Y	YES	0.18
40+	9m	3Y	YES	0.20
50+	3m	2Y	YES	0.20

**Table 5:** best results when considering six-weeks forecasting horizon. All of them run with GBRT algorithm.

FIRST BEST				
Dataset	Feature set	Records time span	Target transform	Model error ratio
0+	12m	4Y	YES	0.15
10+	3m	4Y	NO	0.17
20+	12m	4Y	YES	0.11
30+	12m	4Y	YES	0.19
40+	12m	4Y	NO	0.19
50+	9m	4Y	NO	0.17

**Table 6:** best results when considering three-days forecasting horizon. All of them run with GBRT algorithm.



#### 4.4 Comparison against models from literature

In literature most cited models were naïve, ARIMA, MLR and SVR, being SVR the one with state of art results. We trained MLR and SVR under same conditions as our best models, to understand how compare against them. For a six-weeks horizon, we observed that GBRT outperformed them in all cases. Results are consistent with descriptions from literature, where ARIMA estimates are better than the naïve forecast, but surpassed by the SVR model in all cases. SVR and MLR consistently displayed best results with features computed over last three months regardless of dataset pruning, but MLR shows a rapid prediction quality degradation on the rest of feature sets. Despite this, best results were delivered by MLR over SVR. Results for three-days horizon were similar. MLR had worst results when using 9m or 12m feature sets, followed by naïve forecasting. MLR and SVR displayed best results for 3m and 6m feature sets with MLR beating SVR with a 3m feature set. All GBRT models outperformed MLR and SVR, achieving best performance when features and target are transformed but without dataset pruning.

FIRST BEST					
Dataset	Naive	ARIMA	Best MLR	Best SVR	Best GBRT
0+	-2.03	-1.70	-0.13	-0.86	0.16
10+	-2.04	-1.70	-0.41	-1.01	0.22
20+	-2.07	-1.73	-0.52	-1.13	0.17
30+	-2.08	-1.74	-0.35	-0.97	0.18
40+	-2.08	-1.74	-0.25	-0.87	0.20
50+	-2.02	-1.89	-0.28	-0.57	0.20

Table 7: best models against naïve, ARIMA, MLR and SVR, considering six-weeks horizon.

FIRST BEST					
Dataset	Naive	ARIMA	Best MLR	Best SVR	Best GBRT
0+	-1.73	-1.50	-0.22	-0.76	0.15
10+	-1.73	-1.50	-0.35	-1.01	0.17
20+	-1.77	-1.54	-0.54	-1.17	0.11
30+	-1.77	-1.53	-0.54	-1.08	0.19
40+	-1.77	-1.53	-0.49	-1.03	0.19
50+	-0.90	-1.49	-0.49	-0.47	0.17

Table 8: best models against naïve, ARIMA, MLR and SVR, considering three-days horizon.

#### 4.5 Features contribution

We also explored how much do specific features contribute to predictions, comparing results obtained for best model to those that only take into account historical values of demand records, annual forecasts or open sales. Best results were obtained with demand history features with an average error of at most 8% greater than from models considering all features, with little variation among those trained for either time horizon. Models considering annual sales forecast (Model AF) had an error of 1.85 times the error of the best model on average, while models based only on future sales (Model FS) had greater error averaging 2.18 times that of the best models. We conclude the most important feature is demand history, while the rest of the features contribute to enhance results.

FORECAST 6 WEEKS - FIRST BEST				
Dataset	Model	Model AF	Model FS	Model demand
0+	GBRT-9m-wTT-4Y	1.80	2.24	1.07
10+	GBRT-12m-wTT-3Y	2.01	2.41	1.16
20+	GBRT-12m-wTT-2Y	1.80	2.09	1.15
30+	GBRT-9m-wTT-3Y	1.89	2.20	1.07
40+	GBRT-9m-wTT-3Y	1.89	2.30	1.02
50+	GBRT-3m-wTT-2Y	1.73	1.83	0.96
Average		1.85	2.18	1.07

Table 9: comparison of results with feature sub-sets considering six-weeks horizon.

FORECAST 3 DAYS - FIRST BEST				
Dataset	Model	Model AF	Model FS	Model demand
0+	GBRT-12m-wTT-4Y	1.84	2.43	1.07
10+	GBRT-3m-wTT-4Y	1.82	1.80	1.25
20+	GBRT-12m-wTT-4Y	1.83	2.40	1.02
30+	GBRT-12m-wTT-4Y	1.83	2.59	1.07
40+	GBRT-12m-wTT-4Y	1.69	2.05	1.02
50+	GBRT-9m-wTT-4Y	1.77	1.81	1.06
Average		1.80	2.18	1.08

Table 10: comparison of results with feature sub-sets considering three-days horizon.

#### 4.6 R2 FOR BEST MODELS

After performing the experiments, we computed R2 scores to understand how much variance in the forecasted demand is explained by variables taken into account when performing the prediction. When comparing scores obtained for our best models against those from predictions made by logisticians, we found that our models achieve better scores here too by an average of three to six centesimal points.

### 5 CONCLUSION AND FUTURE WORK

Best models result in an improvement of 10% to 20% over logisticians predictions for both prediction horizons. There is a smaller gap on the three-day prediction horizon, where both predictions are closer to each other. In general, we observe an improvement in results when considering a higher demand history points density. This is also consistent with results regarding features relative importance.

GBRT consistently displays best performance for both forecasting horizons. Regarding feature sets, we observe most models perform best with features computed in a twelve- or nine-months window. When looking for models for six week forecasting horizon, pruning the dataset to a total of three years was optimal, but degraded results for three days horizon.

In the future we would like to enrich existing datasets with time series embeddings as well as products metadata. Time series embeddings should help identify similar timeseries and help make better predictions on products with similar behavior. Products metadata may be used in a similar way, since similar products should have similar demands. Product similarity can be considered from metadata point of view as well as from purchase closeness: items bought together will have similar demands, even though may have different characteristics.

### ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and EU H2020 project FACTLOG under grant agreement No 869951.

### REFERENCES

- [1]- Lee, Hau L., Venkata Padmanabhan, and Seungjin Whang. "Information distortion in a supply chain: the bullwhip effect." *Management science* 43.4 (1997): 546-558.
- [2]- Lee, Hau L., Venkata Padmanabhan, and Seungjin Whang. "Information distortion in a supply chain: the bullwhip effect." *Management science* 43.4 (1997): 546-558.
- [3]- Williams TM (1984). Stock control with sporadic and slow- moving demand. *J Opl Res Soc* 35: 939-948.
- [4]- Syntetos, Aris A., John E. Boylan, and J. D. Croston. "On the categorization of demand patterns." *Journal of the Operational Research Society* 56.5 (2005): 495-503.
- [5]- Matsumoto, Mitsutaka, and Shingo Komatsu. "Demand forecasting for production planning in remanufacturing." *The International Journal of Advanced Manufacturing Technology* 79.1-4 (2015): 161-175.
- [6]- Brühl, Bernhard, et al. "A sales forecast model for the German automobile market based on time series analysis and data mining methods." *Industrial Conference on Data Mining*. Springer, Berlin, Heidelberg, 2009.
- [7]- Spedding, T. A., and K. K. Chan. "Forecasting demand and inventory management using Bayesian time series." *Integrated Manufacturing Systems* 11.5 (2000): 331-339.
- [8]- Liu, Pei, et al. "Application of artificial neural network and SARIMA in portland cement supply chain to forecast demand." 2008 Fourth International Conference on Natural Computation. Vol. 3. IEEE, 2008.
- [9]- Brühl, Bernhard, et al. "A sales forecast model for the German automobile market based on time series analysis and data mining methods." *Industrial Conference on Data Mining*. Springer, Berlin, Heidelberg, 2009.
- [10]- Spedding, T. A., and K. K. Chan. "Forecasting demand and inventory management using Bayesian time series." *Integrated Manufacturing Systems* 11.5 (2000): 331-339.
- [11]- Brühl, Bernhard, et al. "A sales forecast model for the German automobile market based on time series analysis and data mining methods." *Industrial Conference on Data Mining*. Springer, Berlin, Heidelberg, 2009.
- [12]- Dwivedi, Alekh, Maheshwari Niranjan, and Kalicharan Sahu. "A business intelligence technique for forecasting the automobile sales using Adaptive Intelligent Systems (ANFIS and ANN)." *International Journal of Computer Applications* 74.9 (2013).
- [13]- Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016.
- [14]- Yeo, In-Kwon and Johnson, Richard (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87, 954-959.

# Empirical study on the performance of Neuro Evolution of Augmenting Topologies (NEAT)

**Domen Vake**  
domen.vake@innroenew.eu  
Innroenew CoE  
Izola, Slovenia

**Aleksandar Tošić**  
aleksandar.tosic@upr.si  
University of Primorska, UP FAMNIT  
Koper, Slovenia  
Innroenew CoE  
Izola, Slovenia

**Jernej Vičič**  
jernej.vicic@upr.si  
University of Primorska, UP FAMNIT  
Koper, Slovenia  
ZRC-SAZU  
Ljubljana, Slovenia

## ABSTRACT

In this paper we provide empirical results on training a neural network with a genetic algorithm. We test various features of the generalized genetic algorithms, namely speciation and fitness sharing and present the statistical analysis of all three variations. An obstacle avoidance problem was created in which the objective is for vehicles to traverse the course. We present interesting observations about the differences between evolutionary techniques and argue that there is a significant benefit in approaches that aim to diversify the gene pool as a mechanism for avoiding local minima.

I.2.1 ARTIFICIAL INTELLIGENCE Applications and Expert Systems

## 1 INTRODUCTION

Genetic algorithms (GA) have been used extensively for various optimization problems. Arguably, their wide usage spectrum can be accredited to their simplicity and the fact no assumptions are made about the problem. Consequently, most variations of genetic algorithms have strived to maintain these properties. Many techniques were proposed in an effort to diversify the gene pool and at the same time avoid getting stuck in local minima.

In some cases, finding multiple sub-optimal solutions is beneficial [4]. With the recent development of neural networks, genetic algorithms have regained a lot of attention as a viable learning technique. There are many variations to how typical GA functions such as gene encoding, crossover, and mutation are implemented when applied to neural networks. A very promising family of algorithms are descendents of the general NEAT algorithm.

Authors proposed some extensions of the original NEAT algorithm [10] such as rtNEAT [8] that allows evolution to occur in real time rather than through the iteration of generations as used by most genetic algorithms. The basic idea is to put the population under constant evaluation with a "lifetime" timer on each individual in the population. Phased pruning implemented in SharpNEAT framework [2] adds periodic

pruning of the network topologies of candidate solutions during the evolution process. HyperNEAT [9] is specialized to evolve large scale structures. HyperNEAT has recently been extended to also evolve plastic Artificial Neural Networks and to evolve the location of every neuron in the network separately.

The first video game to implement Content-Generating NEAT (cgNEAT) [3] that evolves custom video game content based on user preferences is Galactic Arms Race, a space-shooter game in which unique particle system weapons are evolved based on player usage statistics. Neuro-Evolving Robotic Operatives (NERO) [5] is a video game that applies NEAT to train robots that compete among themselves. odNEAT [7] is an online and decentralized version of NEAT designed for multi-robot systems, it is executed onboard robots themselves during task execution to continuously optimize the parameters and the topology of the artificial neural network-based controllers.

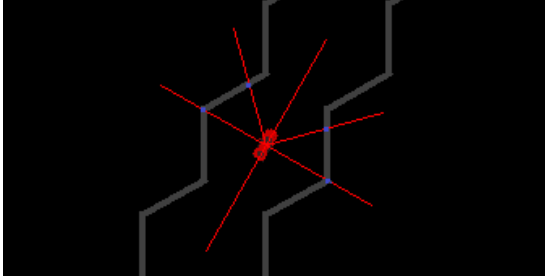
## 2 IMPLEMENTATION

In this section we show specifics to our implementation. The optimization problem is agent based, two-dimensional driving simulation where the objective is to have agents traverse the obstacle course. The agent is considered evolved if it has made a full loop around the track without hitting the wall. It's fitness is based on the distance traveled within a fixed amount of time and is weighted by the amount of checkpoints reached. This ensures the agents move through course and avoid driving in circles.

### Agents

Every tick of simulation the agent's task is to make a decision on the move it wants to perform within the environment. The decision pool consists of five possible moves, which are *do nothing*, *drive forward*, *drive backwards/brake*, *turn left* and *turn right*. The turning is dependant on the speed of the agent, so if the agent is standing still, turning has no effect on it. The decision is chosen with the use of the artificial neural network that is represented as agent's genome and it is based on the agents relative location within the track.

The agents are aware of their surroundings with the help of sight lines, which are represented as lines that fan out from the agent's location as shown in the figure 1. Each line calculates a possible intersection with a wall and tells the agent the distance to the closest wall in the line's direction, if one exists. With agent's speed values are then passed to the genome for a prediction.



**Figure 1:** Figure shows the agent and his sight lines (red lines) and the detection of the wall on the track (blue points).

### Genome encoding

Artificial neural network in the genome is represented as the list of nodes and a list of connections of the network as described in [10]. Upon their creation, all new genes are given an innovation number to ensure the differentiation between the genes. If the gene already exists somewhere in the population (i.e. connection that connects nodes  $x$  and  $y$ ), it is given the same number as the original gene, otherwise a new incrementally higher innovation number is given to the gene. To ensure, that the same gene does not get more than one innovation number, genes must never be deleted, so the number doesn't get lost. For that purpose every connection has a value that represents whether the gene is active and should be represented in the phenotype of the agent. The starting artificial neural network of the agents is a fully connected network with the input layer of size 7 (speed and all sight lines) and output layer of size 5 (all the possible actions of the agents). Activation function of the nodes in the output layer is the *rectified linear unit* and all the other nodes use the *sigmoid* function.

### Crossover

Innovation numbers provide a way to crossover two genomes by matching the genes of the two parents. The genes are split into groups of matching genes (genes that are contained in genomes of both parents), disjoint genes (genes that are contained in only one parent in the middle) and excess genes (genes that are contained in only one parent at the end). When creating an offspring genome, the matching genes are inherited from a random parent, whereas the disjoint and excess genes are inherited from the more fit parent[10].

### Mutation

With crossover the population is likely to discard genes that don't provide good agent behaviour, but that can lead to gene deprivation. We introduce genetic innovation to the system by mutating the genome, where some mutations affect the topology and some the optimization of the network [11].

*Edge mutation.* When optimizing the current topology of the network the existing connections are being mutated. The weight of the connection is either being multiplied by a number between 0 and 2, to provide weight optimization or it is multiplied by -1, to change the polarity of the connection. In case of topological mutation we add a new connection between two unconnected nodes in a way, that doesn't create a cycle in a digraph representation of the network and we give it a random weight or we deactivate a connection. The mutation rates we used were 0.25 for *new connection/deactivate connection*, 0.8 for *adjust weight* mutation and 0.2 for *flip weight* mutation.

*Vertex mutation.* We can also mutate the genome of the network by adding new vertices to increase its complexity and add new options to the pool of solutions. A new vertex is added by choosing a random edge  $e$  between vertices  $A$  and  $B$  and deactivating it. A new vertex  $C$  is inserted, and two new edges created connecting vertex  $A$  and  $B$  with  $C$ . The weight of the edge leading to the new vertex  $C$ , is set to 1 and the edge leading to the vertex  $B$  is set to same weight of the disabled edge  $e$  to minimize the performance impact of the new genes on the genome. The mutation rates for the *new node* mutation that we used were 0.01.

### Species

When adding innovation to the genome, it is likely that the mutation will first reduce the fitness of the agent and will be removed before it has the chance to evolve and optimize. To counter that and protect innovation, we introduce the notion of speciation, where the agents are split into groups that represent species, based on the genotypes.

*Genetic Distance.* To find the genetic difference between two genomes the idea of compatibility distance ( $\delta$ ) is introduced. The less genetic history two genomes have, the more disjoint ( $D$ ) and excess genes ( $E$ ) and the less matching genes they have[10]. These numbers can be normalized with the total number of genes in the larger genome ( $N$ ) and use them to calculate the compatibility distance. We also take into account the average weight differences of matching genes ( $\bar{W}$ ). The coefficients  $w_1$ ,  $w_2$  and  $w_3$  represent the importance of each of the factors and can be adjusted.

$$\delta = \frac{w_1 E}{N} + \frac{w_2 D}{N} + w_3 \bar{W}$$

Every generation each agent is placed into a species, if he's compatibility distance to the species representative is smaller than the prefixed threshold ( $\delta_t$ ). If the agent does not fit into any of the species, he now represents a new species. The weights that were used in our case are  $w_1 = 1.3$ ,  $w_2 = 1.3$ ,  $w_3 = 1.0$  and  $\delta_t = 2.0$ .

**Selection.** In the selection step, we remove the worse half of the agents from the population based on their fitness score. That would in general remove most of the innovation within the population, since the mutated agents tend to perform worse. So instead of removing bottom half in general, we do it per species. That means that every agent only competes with agents that are a part of the same species. This provides an extra layer of protection of the new genes that have not yet had the time to adjust and optimize.

**Explicit Fitness Sharing.** We used the *explicit fitness sharing*[1] niching technique, which normalizes the fitness ( $f$ ) of the agent according to the size of the species that he's in. With niching it is unlikely that one species would take over the whole population therefore it widens the search in the solution space.

$$f'_i = \frac{f_i}{\sum_{j=1}^n sh(\delta(i, j))}$$

If the distance between the agents  $i$  and  $j$  ( $\delta(i, j)$ ) is smaller than the threshold, the value of the sharing function  $sh$  is set to 1 otherwise its set to 0[6].

### 3 RESULTS AND CONCLUSIONS

Four tracks were prepared as shown on figure 4. We tested how the model performs if we remove the different features of the algorithm that provide the protection of the innovation within the population. Explicit fitness sharing and speciation were chosen. From that, we formed four tests: *normal*, that includes all sections of the algorithm, *no efs*, that have explicit fitness sharing disabled, *no speciation* with disabled speciation (only one species during simulation) and *no efs and speciation* that does neither include the explicit fitness sharing nor speciation. The size of the population was set to 1000 and all simulations ran for 250 generations whereas every generation was 750 ticks long. Each test was ran ten times on each of the tracks and for every generation max fitness, mean fitness, standard deviation and whether the model has found the sufficient solution were collected. Also every fifth generation we collected the data of all species, their size, max fitness, mean fitness and standard deviation.

Table 1 shows the aggregated results for individual tracks across multiple runs. Track 1 has seen the best results and least complexity in the neural network as agents only need to turn one direction to successfully traverse the track. The

second best results were obtained by track 3, in which 75% of the simulations evolved and completed the track. Tracks 2 and 4 were arguably the hardest with 60% and 40% respectively. When considering only the evolution rate, all the test showed similar results, since all tests had 67.5-70.0% evolution rate. The test, where explicit fitness sharing was disabled found the best solution in two of the four tracks. This shows, that the weights for the explicit fitness sharing might not have been optimal, for the problem at hand. In the column  $\mu(x)$  shows the mean fitness through all the generations and simulations of the test. The data shows, that the test where speciation and explicit fitness sharing were both disabled in general performed better. However the optimal solution was not found in any of the tracks. This could be a consequence of the model finding and optimizing to a local minimum fast, but due to the lack of innovation, being unable to escape.

**Table 1: Table shows the data gathered from tests. Every row represents summarized data from 10 iterations of the test on specific track Fitness is represented as  $x$  and is normalized by the highest fitness achieved on that track. Tests: N-normal, E-no efs, S-no speciation, SE-no efs and speciation**

	Track	Test	$max(x)$	$\mu(x)$	$\sigma(x)$	Evolved[%]
1	Track 1	N	0,99	0,12	0,58	100
2	Track 1	E	0,98	0,18	0,45	100
3	Track 1	S	1	0,13	0,57	100
4	Track 1	SE	0,98	0,21	0,46	100
5	Track 2	N	0,96	0,03	0,15	50
6	Track 2	E	1	0,05	0,20	50
7	Track 2	S	0,99	0,04	0,24	80
8	Track 2	SE	0,88	0,05	0,19	60
9	Track 3	N	0,98	0,05	0,24	80
10	Track 3	E	1	0,12	0,37	80
11	Track 3	S	0,93	0,06	0,26	80
12	Track 3	SE	0,83	0,09	0,28	60
13	Track 4	N	1	0,01	0,16	40
14	Track 4	E	0,61	0,03	0,14	50
15	Track 4	S	0,64	0,01	0,05	20
16	Track 4	SE	0,84	0,06	0,24	50

Figures 2, and 3 illustrate the impact of fitness sharing on the evolution of existing species and the emergence of new ones. We observe that the number of different species that emerged is significantly higher when fitness sharing is enabled. This is expected as the mechanism allows new species to be preserved across generations in order to diversify the gene pool. However, we also observe that a significant number of species that emerged survived across all generations and achieved significant improvements to their fitness (representative). Additionally, we can observe that in both cases,

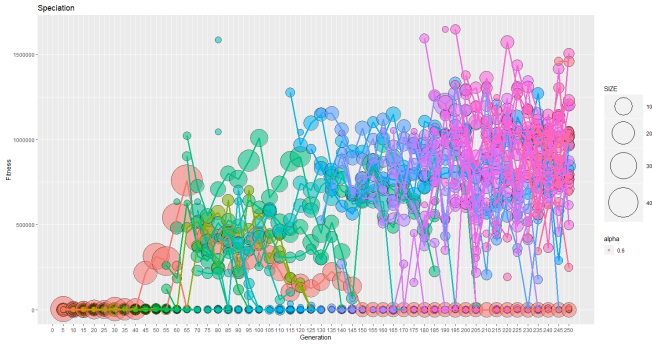


Figure 2: Species with explicit fitness sharing enabled

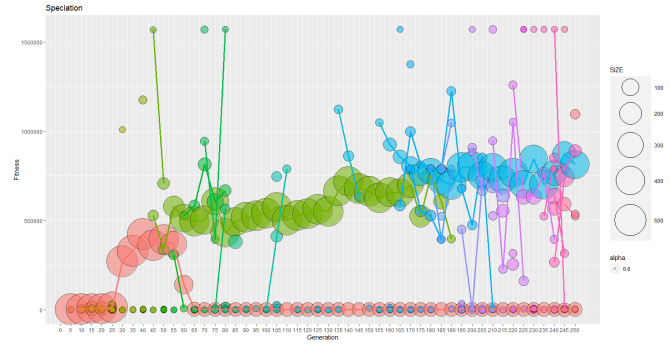


Figure 3: Species with explicit fitness sharing disabled

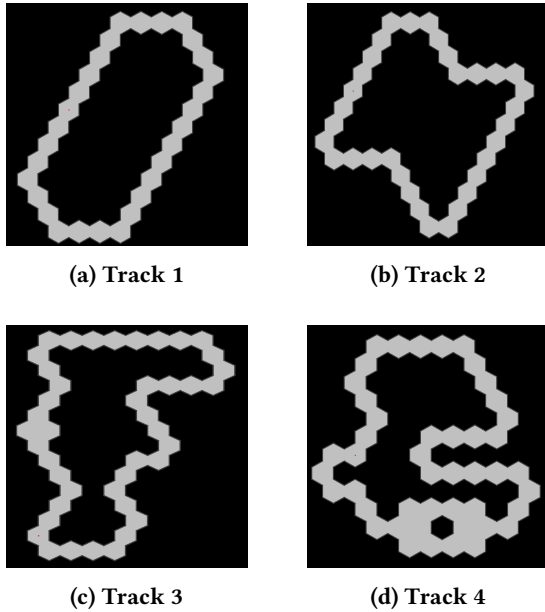


Figure 4: Tracks used for testing the model's performance

some newly emerged species never evolve and improve their fitness which eventually causes them to be removed. This indicates that even with explicit fitness sharing, species with bad genes do not impact the overall results even though their genes are initially protected.

With no fitness sharing, there is a trend of one big species and a few smaller ones, that explore the solution space and when a new innovation is found with better results, it takes over the population. Contrary to that, evolution of the population, that shares fitness splits into more species and is searches the space for a wider set of solutions.

All the presented software is available under opensource licence at Github<sup>1</sup>.

<sup>1</sup>NEAT-driving: <https://github.com/VakeDomen/NEAT-driving>

## 4 ACKNOWLEDGMENTS

The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE (Grant Agreement #739574) under the H2020 Widespread Teaming programme and investment funding from the Republic of Slovenia and the European Regional Development Fund.

## REFERENCES

- [1] David E Goldberg, Jon Richardson, et al. 1987. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 41–49.
- [2] Colin Green. 2004. Phased searching with NEAT: alternating between complexification and simplification. *Unpublished manuscript* (2004).
- [3] Erin J Hastings, Ratan K Guha, and Kenneth O Stanley. 2009. Evolving content in the galactic arms race video game. In *IEEE CIG*. 241–248.
- [4] Jian-Ping Li, Marton E Balazs, Geoffrey T Parks, and P John Clarkson. 2002. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary computation* 10, 3 (2002), 207–234.
- [5] Risto Miikkulainen, Bobby D Bryant, Ryan Cornelius, Igor V Karpov, Kenneth O Stanley, and Chern Han Yong. 2006. Computational intelligence in games. *Computational Intelligence: Principles and Practice* (2006), 155–191.
- [6] Bruno Sareni and Laurent Krahenbuhl. 1998. Fitness sharing and niching methods revisited. *IEEE transactions on Evolutionary Computation* 2, 3 (1998), 97–106.
- [7] Fernando Silva, Paulo Urbano, Luis Correia, and Anders Lyhne Christensen. 2015. odNEAT: An algorithm for decentralised online evolution of robotic controllers. *Evolutionary Computation* 23, 3 (2015), 421–449.
- [8] Kenneth O Stanley, Bobby D Bryant, and Risto Miikkulainen. 2003. Evolving adaptive neural networks with and without adaptive synapses. In *CEC'03*, Vol. 4. IEEE, 2557–2564.
- [9] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. 2009. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life* 15, 2 (2009), 185–212.
- [10] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.
- [11] Pushpendra Kumar Yadav and NL Prajapati. 2012. An overview of genetic algorithm and modeling. *IJSRP* 2, 9 (2012), 1–4.

# Learning Hand-Eye Coordination on NAO and its Applications

Ana Gaja Boc

Jožef Stefan Institute, Department of Knowledge  
Technologies  
Jamova 39, 1000 Ljubljana  
Fakulteta za računalništvo in informatiko  
Večna pot 113, 1000 Ljubljana  
ab9870@student.uni-lj.si

Sara Bertoneclj Čadež

Jožef Stefan Institute, Department of Knowledge  
Technologies  
Jamova 39, 1000 Ljubljana  
Fakulteta za računalništvo in informatiko  
Večna pot 113, 1000 Ljubljana  
sb4914@student.uni-lj.si

## ABSTRACT

This paper focuses on learning hand-eye coordination on robot NAO. It elaborates on two different approaches for computing inverse kinematics using neural networks. It also presents two applications, based on the computed inverse kinematics: a system that enables the robot to play tic-tac-toe against a human opponent and a system that enables the robot to replicate simple shapes that it sees.

## Keywords

robotics, inverse kinematics, vision recognition

## 1. INTRODUCTION

Inverse kinematics is commonly used for solving problems such as object grasping, visually guided tasks and also in 3D animation for interaction between characters and other objects in the animated world. While calculating the forward kinematics, that is the position of the end effector based on joint configuration, is a fairly easy problem to solve, inverse kinematics proves to be more challenging because of its multiple solutions.

Traditional methods are computationally expensive, because they rely on constructing and operating on large and complex matrices. Such is the iterative method, which requires the inversion of the Jacobian matrix. There are also alternative solutions that do not require matrices or rotational angles, such as FABRIK [1] (Forward and Backward Reaching Inverse Kinematics). This heuristic algorithm performs simple, iterative operations that gradually lead to an approximation of the solution, by finding the joint coordinates as being points on a line. Inverse kinematics for the NAO robot implemented with the FABRIK algorithm were described by Renzo Poddighe [5], in an article which focuses on a system that enables the robot to play tic-tac-toe, very much like one of the applications presented in this paper. We propose a third approach by calculating the inverse kinematic with neural networks.

## 2. NAO ROBOT

The first public version of robot NAO was presented in March 2008. Since then six versions of this humanoid have been produced, each having better cameras, CPU, speech

synthesis in more languages and better face recognition. For work described in this paper we used NAO version 4.

It has 25 degrees of freedom. The motion ranges of two joints are important for the computation of inverse kinematics: the right shoulder roll which has the motion range from -76 to 18 degrees and the right elbow roll which has the motion range from 0 to 88.5 degrees. It has 1.6 GHz CPU ATOM Z530, 1 GB of RAM and 2 GB of Flash memory. The camera has up to 1280x960 resolution with 60.9 degrees horizontal field of view.

NAO's operating system is based on Linux Gentoo and named NAOqi OS. It has built-in libraries that are needed for the NAOqi Framework, the main software that allows communication between the different modules, programming and information sharing.

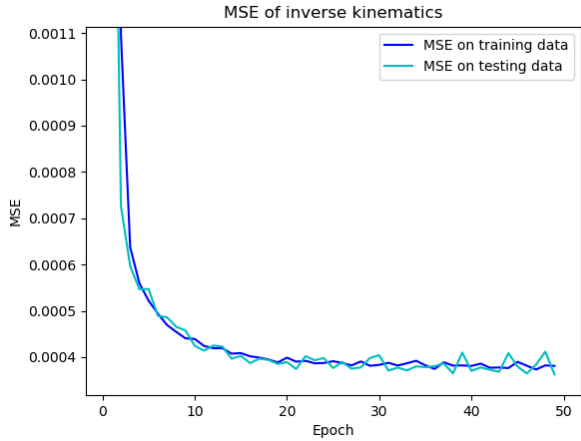
## 3. INVERSE KINEMATICS

Inverse kinematics was calculated with two different approaches for two different implementations. For the game of tic-tac-toe joint positions were calculated for pixels on the image of the gaming surface taken with the robot's camera. For drawing simple shapes joint positions were calculated for x and y coordinates of points on the tablet.

### 3.1 Neural networks

In both cases inverse kinematics was calculated using a regression neural network. For playing the game of tic-tac-toe, the angles in NAO's arm were measured, by tracking a red pen, while the robot moved it across the gaming surface. Recorded data consisted of pixel coordinates of the tip of the red pen in the image taken by the robot's camera and the shoulder and elbow roll angles. For drawing, the shoulder roll, shoulder yaw, elbow roll and elbow yaw were measured, using a graphic tablet and a stylus pen. While holding the pen the robot's hand was moved around on the tablet surface. As the robot was moving the pen, a program recorded the angle of each of the aforementioned joint and the position of the cursor. There were 279 training samples collected, for playing the game of tic-tac-toe and 10000 training samples for drawing simple shapes. With the second method of recording data we could gather a much more extensive sample size. We could not use the same method for playing tic-tac-toe because the model had to use the position in coordinate system of NAO's camera allowing the program





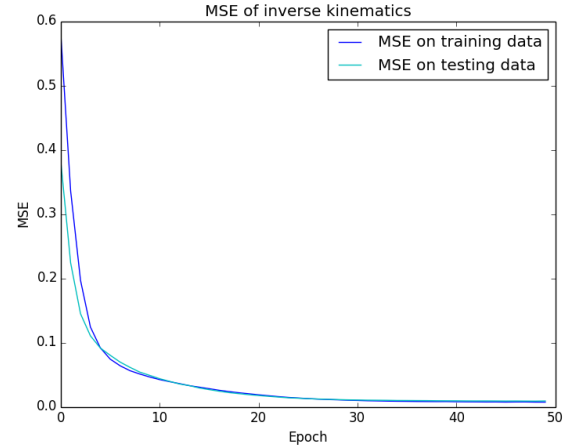
**Figure 1: MSE of the neural network for learning inverse kinematics for drawing simple shapes.**

to work regardless of where in NAO’s field of view the playing area was located. Meanwhile the model for drawing was transforming the coordinates from the picture taken with the camera onto a fixed surface in front of the robot. With fewer training samples the model worked better using just two degrees of freedom, meanwhile the model with larger sample size worked better with four degrees of freedom.

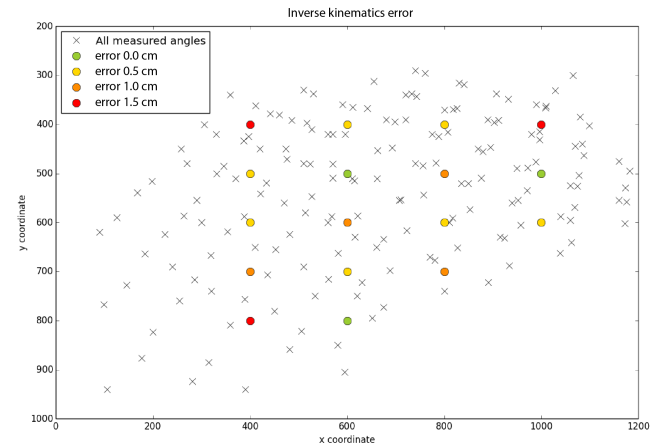
Inverse kinematics was calculated using a simple neural network. The neural network was implemented with Keras sequential model. Input variables are x and y coordinates. While neural network for tic-tac-toe had only one hidden layer, neural network for drawing had two identical hidden layers. They had 32 nodes and rectified linear unit activation function. The output layer had two/four nodes which correspond to the dimensions of the output variable (array of two/four angles in radians). To evaluate weights we used the mean square error loss function that calculates the mean error of both/all four angles and the efficient stochastic gradient descent algorithm Adam [3] for optimisation. To train the model we used 50 epochs and a batch size of 10 for the smaller neural network and 50 for the bigger neural network. The mean square error of the final model for drawing was  $4.2 \times 10^{-4}$ , the error at each epoch is shown in Figure 1. Mean square error of tic-tac-toe model was  $6.2 \times 10^{-3}$ , the error at each epoch is shown in Figure 2.

We also calculated inverse kinematics with Support Vector Regression. With training samples for drawing, the mean square error was  $1.1 \times 10^{-3}$ , which is considerably worse than  $4.2 \times 10^{-4}$  error obtained using neural network. With training samples for tic-tac-toe mean square error was  $8.1 \times 10^{-3}$ , while neural network error was  $6.2 \times 10^{-3}$ .

Because drawing requires higher precision, there were more samples collected and a bigger neural network built. It is also because of the large number of samples, that we get higher accuracy by predicting four and not just two angles. For playing tic-tac-toe, precision up to 1 cm is adequate and it can be achieved by predicting just two angles on a smaller data set. Measured precision of inverse kinematics is shown



**Figure 2: MSE of the neural network for learning inverse kinematics for playing tic-tac-toe.**



**Figure 3: The pixels for which the corresponding arm angles were measured (crosses) and the error of the computed inverse kinematics (dots).**

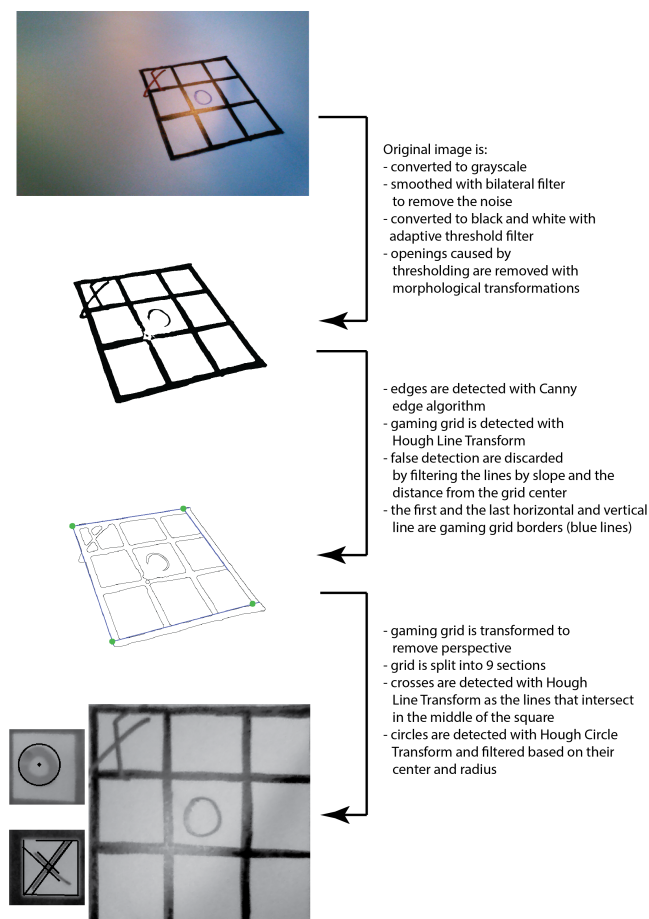
in the Figure 3.

## 4. APPLICATIONS

We developed two different applications for inverse kinematics. The first one enables NAO to play tic-tac-toe, a simple game where two players take turns in placing their mark (cross or circle) on a grid of size  $3 \times 3$ . The player that first succeeds in placing three of his marks horizontally, vertically or diagonally, wins the game. The second one focuses on NAO drawing solid simple shapes, which it captures with its camera.

### 4.1 Tic-tac-toe

To solve the problem of the robot playing tic-tac-toe, two additional separate modules were developed. A vision recognition module was developed, for recognising the location of the gaming grid and current state of the game. A strategy module was implemented, for deciding which move will most likely lead the robot to victory.



**Figure 4: Recognising the state of the game.**

The vision recognition component is written in Python using the OpenCV library. Before each move, the robot takes a picture on which the state of the game and the location of gaming board are detected.

The image processing pipeline is shown in Figure 4. Probabilistic Hough Line Transform [4], which returns an array of the start and the end points of all the detected lines is used for the gaming grid detection. These lines are then separated into horizontal and vertical lines and ordered by their position in the image. If there are more than four horizontal or vertical lines detected, they are filtered by their slope and the distances from the previous and the next line. The lines that deviate the most are discarded as false detections. After that, we can be sure that the first and the last horizontal and the first and the last vertical line are the borders of our gaming grid. The intersections of these four lines are also calculated and those four points are used to do a warp transform of the gaming grid, so that the camera perspective is removed and the grid is seen as from a vertical position.

The gaming grid is then split into thirds vertically and horizontally, which gives us nine fields on which there could be a circle or a cross. Hough Circle Transform [7] is used, for circle detection. If a circle is found, its radius and centre

are compared to the expected values. Hough Line Transform is used, for cross detection. If there are lines found, possible intersections of these lines are compared to the expected values. If there is an intersection in the middle of the field, then a cross is detected.

In the experiments, the state of the game is correctly recognised in 39/40 cases, which is 97,5% success rate. In 1/40 cases there is an error in vision recognition because of falsely discarding one or more lines as false detections.

The second component of the system is an algorithm that chooses the next move, based on the current game state. Minmax decision rule with alpha-beta pruning was chosen for that, which makes the robot unbeatable at tic-tac-toe.

When the current state of the game is recognised in the image, that information is passed to the algorithm for choosing the next move. Inverse kinematics is then calculated, based on the location of the chosen move. Four arm positions corresponding to the field vertexes are calculated, which are then used for drawing the cross, by connecting the opposite vertexes of the field.

## 4.2 Drawing simple shapes

In this application our goal is to teach NAO to draw a simple shape that it sees through its camera. Besides calculating the inverse kinematics, another problem we face is computing the points (in the correct order) that the robot must reach to render the shape it was shown. To solve this we use computer vision to process an image that was captured by the robots camera.

When the robot is ready to draw, it will wait to be presented with an image. For the following algorithm to work the image must be of a solid shape on a single-coloured background. When the image is in the field of view of the robot's camera we capture it by pressing on one of the tactile buttons on its head. After the image is captured we need to process it with OpenCV library for Python in order to extract the contours. First the image must be converted from colour image to grayscale. Then a bilateral filter is used to reduce the noise while maintaining defined edges. On the filtered image we can then use canny edge detection [2] to find the edges of the shape we want the robot to draw. From the edge image we can then extract the contours as seen in Figure 5.

When extracting the contours we chose to store all the points along the boundary by not using any chain approximation. This makes the drawing process very slow but it is the most accurate for all types of shapes. Choosing simple or Teh-Chin chain approximation [6] works for curved lines but it can cause problems when drawing shapes with long straight lines. It reduces the number of stored points to mostly just the corner points which means the angles for connecting the points have to be interpolated over a relatively long distance. The angle interpolation of vertical lines produces jagged lines as seen in Figure 6.

After we have extracted the contours we use bounding rectangle to determine the height and width of the shape so we can scale it to fit the robot's drawing area. Once all the

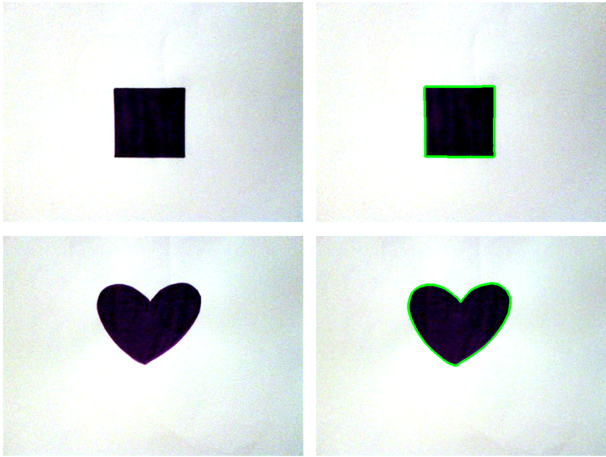


Figure 5: Two pictures NAO captured (on the left) and with extracted contours (on the right).

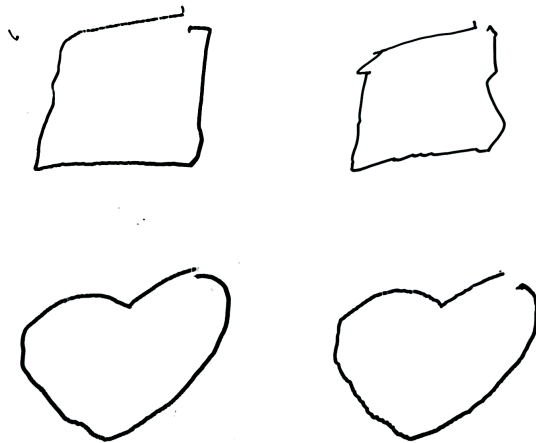


Figure 6: Two drawings produced without chain approximation (on the left) in comparison with two drawings produced with Teh-Chin chain approximation (on the right).

points in the contours are properly scaled we can calculate the angles of robot's joints using the model from the previous section.

Because of the friction between the pen and the drawing surface as well as because of the slight looseness of NAO's joints very small errors accumulate while the robot is drawing. This results in the contours on the render images not being connected at the ends, as shown in Figure 6.

## 5. CONCLUSIONS

In this paper we solved the problem of hand-eye coordination using neural networks and applied it to two real life problems.

The first system is able to play tic-tac-toe game against a human opponent without losing a single game. Inverse kinematics is precise enough so that the cross that robot draws always has a centre inside the selected field on the gaming grid. Vision recognition correctly recognises the state of the game in 97% of cases. Currently NAO can only draw crosses. We wish to develop the system further so that it will be able to draw circles too. For drawing crosses, four angles that correspond to field vertexes need to be calculated. If the robot were to draw circles, we would need to calculate inverse kinematics for a few dozen positions corresponding to the circle on the field that the robot would draw. The precision of inverse kinematics would also need to be much higher.

The second system successfully extracts contours of solid shapes of one colour and replicates them. By making the image processing more robust we could generalise it to work for more complex drawings. We also wish to improve the processing by combining the current method of contour extraction with other methods of line detection so that the system will be able to draw simple lines that are not joined at the ends.

## 6. REFERENCES

- [1] A. Aristidou and J. Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011.
- [2] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR 2014*, 2014.
- [4] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *CVIU*, 78(1):119–137, 2000.
- [5] R. Poddighe. Playing tic-tac-toe with the nao humanoid robot. 2013. url: <https://project.dke.maastrichtuniversity.nl/robotlab/wp-content/uploads/Renzo-Poddighe.pdf> accessed: 18-August-2019.
- [6] C. Teh and R. Chin. On the detection of dominant points on digital curve. *PAMI*, 11(8):859–872, 1989.
- [7] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler. Comparative study of hough transform methods for circle finding. *Image Vision Comput.*, 8(11):71–77, 1990.

## Indeks avtorjev / Author index

Belayeva E. ....	21
Bertoncelj Čadež Sara .....	65
Bizjak Luka .....	25
Boc Ana Gaja .....	65
Brezec Sara.....	17
Čerin Matej.....	45
Fortuna Blaž .....	57
Fuart F. ....	21, 33
Grobelnik Marko .....	9, 21
Hirsch M.....	33
Kavšek Branko .....	41, 49, 53
Kenda Klemen.....	17, 29, 37, 45
Kojanec Patrik .....	53
Koprivec Filip .....	37, 45
Košmerlj Aljaž .....	21, 25
Kralj Samo .....	29
Leban G. ....	21
Malik Omar .....	41
Massri Beshir M. ....	17
Mattiev Jamolbek .....	49
Mexia R. ....	33
Mladenić Dunja.....	5, 9, 21, 41, 57
Mladenić Grobelnik Adrian.....	9
Novak Erik .....	5, 17, 29
Paolotti D. ....	33
Peternelj Jože .....	37
Pita Costa Jao .....	21, 33
Rei L.....	21
Rožanec Jože Martin .....	57
Stopar L. ....	21, 33
Sunar Ayşe Saliha .....	5
Szymanski Bolesław K.....	41
Teixeira César A. D.....	53
Torkar Miha .....	13, 25
Tošić Aleksandar.....	61
Trichilo Giulio.....	13
Urbančič Jasna .....	5
Urbančič Živa.....	29
Vake Domen.....	61
Vičič Jernej.....	61

