Zbornik 24. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2021

Zvezek A

Proceedings of the 24th International Multiconference

INFORMATION SOCIETY – IS 2021

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredniki / Editors

Mitja Luštrek, Matjaž Gams, Rok Piltaver

http://is.ijs.si

8. oktober 2021 / 8 October 2021
Ljubljana, Slovenia

Uredniki:

Mitja Luštrek
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Matjaž Gams
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Rok Piltaver
Outfit7
in Odsek za inteligentne sisteme, Institut »Jožef Stefan«, Ljubljana

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

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

Ljubljana, oktober 2021

Informacijska družba
ISSN 2630-371X

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2021

Štiriindvajseta multikonferenca Informacijska družba (http://is.ijs.si) je preživela probleme zaradi korone v 2020. Odziv se povečuje, v 2021 imamo enajst konferenc, a pravo upanje je za 2022, ko naj bi dovolj velika precepljenost končno omogočila normalno delovanje. Tudi v 2021 gre zahvala za skoraj normalno delovanje konference tistim predsednikom konferenc, ki so kljub prvi pandemiji modernega sveta pogumno obdržali visok strokovni nivo.

Stagnacija določenih aktivnosti v 2020 in 2021 pa skoraj v ničemer ni omejila neverjetne rasti IKT, informacijske družbe, umetne inteligence in znanosti nasploh v 2021, ampak nasprotno, rast znanja, računalništva in umetne inteligence se nadaljuje z že kar običajno nesluteno hitrostjo. Po drugi strani pa se je še dodatno pospešil razpad družbenih vrednot, zaupanje v znanost in razvoj, kar se kaže predvsem v raznih proti[illegible]ilnih gibanjih. Žal čedalje več ljudi verjame, da je Zemlja ploščata, da je cepivo za korono škodljivo, da virusov ni. Razkorak med rastočim znanjem in vraževerjem se povečuje tudi v zadnjem letu in zavedanje večine ljudi, da to pelje nazaj v srednji vek, čedalje bolj krepi, kar je bistvena sprememba glede na 2020.

Letos smo v multikonferenco povezali enajst odličnih neodvisnih konferenc, ki ima okoli 150 večinoma spletnih predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic ter [illegible] obiskovalcev. Prireditev so spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad – seveda večinoma preko spleta. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica (http://www.informatica.si/), ki se ponaša s 45-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2021 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Odkrivanje znanja in podatkovna skladišča
- Kognitivna znanost
- Ljudje in okolje
- 50-letnica poučevanja računalništva v slovenskih srednjih šolah
- Delavnica projekta Batman
- Delavnica projekta Insieme Interreg
- Delavnica projekta Urbanite
- Študentska konferenca o računalniškem raziskovanju 2021
- Mednarodna konferenca o prenosu tehnologij
- Vzgoja in izobraževanje v informacijski družbi

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi 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.

Nagrade bodo razglašene v petek, 8.10.2021. Podelili bomo nagrado za življenjske dosežke v čast Donalda Michieja in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe je prejme …. Priznanje za dosežek leta pripada …. Podeljujemo tudi nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono prejme …, jagodo …. Čestitke nagrajencem!

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

FOREWORD - INFORMATION SOCIETY 2021

The 24th Information Society Multiconference (http://is.ijs.si) survived the COVID-19 problems. In 2021, there are eleven conferences with a growing trend and real hopes that 2022 will be [illegible] due to successful vaccination. The multiconference survived due to the conference presidents that bravely decided to continue with their conference despite the first pandemics in the modern era.

The COVID-19 pandemic did not decrease the growth of ICT, information society, artificial intelligence and science overall, quite on the contrary – the progress of computers, knowledge and artificial intelligence continued with the fascinating growth rate. However, COVID-19 did increase the downfall of societal trust in science and progress, most evident in anti-vaccination movements. The number of people believing that the Earth is flat is growing as well as those that believe that the COVID-19 vaccines are harmful or even that viruses don't exist at all. On the other hand, the awareness of the majority population that such approaches c[illegible]d to returning to the Dark Ages, grows to the point that proper actions against this phenomenon are promoted.

The Multiconference is running parallel sessions with 150 presentations of scientific papers at eleven conferences, many round tables, workshops and award ceremonies, and 300 attendees. Selected papers will be published in the Informatica journal with its 45-years tradition of excellent research publishing.

The Information Society 2021 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Data Mining and Data Warehouses
- Cognitive Science
- People and Environment
- 50-years of High-school Computer Education in Slovenia
- Batman Project Workshop
- Insieme Interreg Project Workshop
- URBANITE Project Workshop
- Student Computer Science Research Conference 2021
- International Conference of Transfer of Technologies
- 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.

The awards will be announced [illegible] 8.10.2021. [illegible] award for life-long outstanding contributions will be presented in memory of Donald Michie and Alan Turing. The Michie-Turing award was given to … for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, a recognition for current achievement was awarded to … The information lemon goes to the …, and the information strawberry to the … 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
Mitja Lasič
Blaž Mahnič
Jani Bizjak
Tine Kolenik

Programme Committee

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

KAZALO / TABLE OF CONTENTS

Zbornik 24. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2021

Zvezek A

Proceedings of the 24th International Multiconference

INFORMATION SOCIETY – IS 2021

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredniki / Editors

Mitja Luštrek, Matjaž Gams, Rok Piltaver

http://is.ijs.si

8. oktober 2021 / 8 October 2021
Ljubljana, Slovenia

PREDGOVOR

Po zaslugi pandemije COVID-19 še vedno živimo v bolj zanimivih časih, kot bi si želeli, vendar umetne inteligence to ne moti in napreduje s podobnim tempom kot pretekla leta. Računalniški vid in obdelava naravnega jezika sta še vedno vroči področji, pred nedavnim pa nam je OpenAI postregel s parom navdušujočih kombinacij obojega. Prva je DALL-E, globoka nevronska mreža, izpeljana iz OpenAIjeve slavne mreže za generiranje besedila GPT-3, ki je sposobna »razumeti« opis slike in nato takšno sliko generirati. Pri tem je kos slikam, na kakršne prej ni naletela – generirati zna denimo prav čedno sliko redkve daikon v baletnem krilcu, ki sprehaja psa. Druga, CLIP, deluje obratno in generira besedilne opise slik. Še en viden dosežek zadnjega časa prihaja s področje biologije in medicine, ki sta zelo plodni področji za uporabo umetne inteligence. Algoritem AlphaFold 2, ki – podobno kot večina pomembnih dosežkov umetne inteligence zadnjih let – temelji na globokih nevronskih mrežah, je dosegel dramatičen napredek pri določanju strukture beljakovin, kar je težaven problem, pomemben za razvoj zdravil.

Posebej odmeven nedaven dosežek umetne inteligence iz domačih logov je metoda za priporočanje optimalnih ukrepov zoper COVID-19, ki jo je razvila ekipa Odseka za inteligentne sisteme na Institutu Jožef Stefan. Pri tej sodbi avtorji predgovora sicer nismo povsem nepristranski, saj sva k dosežku dva prispevala, a drugo mesto ne tekmovanju XPrize Pandemic Response Challenge s polmilijonskim nagradnim skladom našo trditev potrjuje. Za uspeh tokrat ni bila potrebna globoka nevronska mreža – metoda kombinira epidemiološki model SEIR, klasično strojno učenje in večkriterijsko optimizacijo z evolucijskim algoritmom. Na Slovenski konferenci o umetni inteligenci je predstavljen le delček tega dela, več o njem pa je moč izvedeti na Delavnici projekta Insieme Interreg, ki prav tako poteka v okviru Informacijske družbe.

Posebej veliko število drugih delavnic in konferenc na Informacijski družbi letos je sicer dobro za multikonferenco kot celoto, našo konferenco pa je bržkone prikrajšalo za kak prispevek. K tej težavi moramo dodati še naveličanost raziskovalne srenje nezmožnosti žive udeležbe na konferencah, tako da smo se morali na koncu zadovoljiti s 13 prispevki. Večino je kot po navadi prispeval Institut Jožef Stefan, dobro je zastopana tudi Fakulteta za računalništvo in informatiko Univerze v Ljubljani, druge ustanove pa žal ne. Kljub temu smo poskrbeli, da so prispevki kakovostni, in smo jih zavrnili več kot pretekla leta. Bomo pa prihodnje leta napeli moči, da privabimo več prispevkov iz širšega nabora ustanov.

FOREWORD

Thanks to the COVID-19 pandemic we still live in more interesting time than we would like, but artificial intelligence is not much bothered by this and is progressing as rapidly as in the recent years. Computer vision and natural language processing are still hot topics, and OpenAI recently provided a pair of exciting combinations of the two. The first is DALL-E, a deep neural network derived from OpenAI's famous language generation network GPT-3. It can »understand« a description of an image and then generate such an image. It can handle images never encountered before – for instance, it can generate a nice image of a daikon radish in a tutu walking a dog. The second is CLIP, which works in reverse and generates descriptions of images. Another prominent recent achievement comes from biology and medicine, which is fruitful ground for applications of artificial intelligence. The AlphaGo 2 algorithm, which – like most main achievements of artificial intelligence in the recent years – is based on deep neural networks, achieved a breakthrough in protein folding. This is a hard problem important for drug discovery.

A prominent recent Slovenian achievement of artificial intelligence is a method for recommending optimal interventions against COVID-19, which was developed by a team from the Department of Intelligence Systems at Jožef Stefan Institute. The authors of this foreword are not entirely unbiased when we say this, because two of us contributed to the achievement, but second placed at the XPrize Pandemic Response Challenge with a prize purse of half a million lends credence to our claim. This success did not require a deep neural network – the method combines a SEIR epidemiological model, classical machine learning and multi-objective optimisation with an evolutionary algorithm. The Slovenian Conference of Artificial Intelligence presents only a small part of this work, while more can be learned in the Insieme Interreg project workshop.

A particularly large number of other workshops and conference at Information Society this year are good for the multi-conference as a whole, but probably deprived our conference for a few papers. Another problem is that the research community is getting tired of the inability to attend conferences live, which is why we ended up with only 13 papers. Most of them, as usual, come from Jožef Stefan Institute. The Faculty of Computer and Information Science of the University of Ljubljana is also well represented, while other institutions less so. Despite this we made sure that the papers are high-quality, and we turned away more than usual. But our goal for the following years is of course to secure more papers from a wider range or institutions.

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Mitja Luštrek

Matjaž Gams

Rok Piltaver

Cene Bavec

Jaro Berce

Marko Bohanec

Marko Bonač

Ivan Bratko

Bojan Cestnik

Aleš Dobnikar

Bogdan Filipič

Borka Jerman Blažič

Marjan Krisper

Marjan Mernik

Biljana Mileva Boshkoska

Vladislav Rajkovič

Niko Schlamberger

Tomaž Seljak

Miha Smolnikar

Peter Stanovnik

Damjan Strnad

Vasja Vehovar

Martin Žnidaršič

Estimating Client's Job-search Process Duration*

Viktor Andonovic[1]
Knowledge Technologies
[1]Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia
viktor.andonovikj@ijs.si

Pavle Boškoski[2]
Knowledge Technologies
[2]Institute Jožef Stefan
Ljubljana, Slovenia
pavle.boskoski@ijs.si

Biljana Mileva Boshkoska[2,3]
Knowledge Technologies
[2]Institute Jožef Stefan
[3] Faculty of information studies
[2]Ljubljana, Slovenia
[3]Novo Mesto, Slovenia
biljana.mileva@ijs.si

ABSTRACT

Modelling the labour market, analysing ways to reduce unemployment, and creating decision support tools are becoming more popular topics with the rise in digital data and computational power. The paper aims to analyse a Machine Learning (ML) approach for estimating the time duration until a job-seeker finds a job, i.e. leaves the Public Employment Service (PES), after the initial entering. The dataset that we use from PES is complex, and there is almost no correlation between most of the features in it, which makes it challenging for modelling. We used statistical analysis and visualisations to understand the problem better and form a basis for further modelling. As a result, we developed several ML models, including basic multivariate linear regression used for performance comparison with other more specifically designed models.

1 INTRODUCTION

The research field of creating tools for supporting the decision-making process for employment services has attracted significant interest lately. One can track such efforts for more than 20 years [1]. Different variants of tools and systems have been developed and implemented with varying success in different countries. PES is willing to move away from the traditional role of servicing the job-seekers and take a more systematic approach by implementing data-driven solutions in their toolbox. Here, the goal is to create a model that uses available data that describes the job-seekers that have entered the PES and outputs the approximate time (in days) needed for the individual to leave the PES as an employed person.

These factors can be assessed either by introducing experts' knowledge or by extracting the corresponding dynamics directly from the available data. What was (or is) available determines how the models are built and their effectiveness.

The biggest issue when dealing with any modelling, for that matter, is the quality of data. Typically models of the labour flow are built on top of statistical surveys [2]. These data sets comprise a series of snapshots of an individual labour force status observed at discrete time points. Such discrete sampling might be with low frequency in order to truly capture the changing dynamics.
Several methods for approaching similar labour market modelling problems have been implemented in other countries. Finland's Statistical profiling tool, introduced in 2007, consists of a simple logit model [3]. It predicts the probability of long-term unemployment and categorises job seekers into two groups, risk or high-risk of long-term unemployment. In 2012 Ireland implemented a PEX (probability of exit) model using data collected on job-seekers who entered the PES as unemployed during 13 weeks [4]. The PEX tool is a probit model for measuring the job-seeker's probability of exiting unemployment in one year.

As a result of our work, we have developed an ML model that can be used in a PES as a part of their decision toolbox, which can serve as a filtering method that prioritises job-seekers and recognises ones who do not necessarily need PES resources and services, as they will get employed soon regardless of the interventions by the organisation.

2 DATA

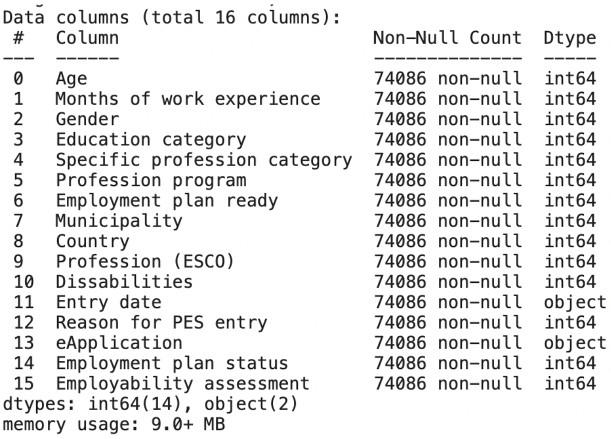
The data used for the paper is provided by a public organisation engaged in the HECAT project [4], which aims at investigation, demonstration and pilotting a profiling tool to support labour market decision making by unemployed citizens and case workers in PES.

2.1 Data Description

The dataset consists of 74086 instances, each representing a client enrolled in the PES, described with 16 sociological, demographic and time-related characteristics, known as features or attributes. The data were obtained during one year. The dataset is complex in a way that its attributes come in a different form (categorical, numerical, date and time), and most of them need to undergo some transformation for the aim of input suitability for different ML models. The general structure of the client's attributes is described by dividing the attributes into several prominent groups: socioeconomic variables (gender, age,

[*] Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2020, 4–8 October 2021, Ljubljana, Slovenia

nationality), information on job readiness (education, health limitations, care responsibilities), and opportunities (regional labour market development), and all available labour market history information, such as prior work experience. Most of the categorical features are given with numbers, where each number represents a unique category, described in a separate CSV file. The target variable is in numeric form, and it is a counter of days that a person stays in the process before exiting the PES. Some of the features in the dataset contain weird values (such as the negative number for clients age), which are a mistake or a result of noise in the data. This indicates the necessity of performing data cleaning and preprocessing before using the dataset to input various ML models. Figure 1 gives an overview of the attributes of the dataset.



```
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Age                           74086 non-null  int64
 1   Months of work experience     74086 non-null  int64
 2   Gender                        74086 non-null  int64
 3   Education category            74086 non-null  int64
 4   Specific profession category  74086 non-null  int64
 5   Profession program            74086 non-null  int64
 6   Employment plan ready         74086 non-null  int64
 7   Municipality                  74086 non-null  int64
 8   Country                       74086 non-null  int64
 9   Profession (ESCO)             74086 non-null  int64
 10  Dissabilities                 74086 non-null  int64
 11  Entry date                    74086 non-null  object
 12  Reason for PES entry          74086 non-null  int64
 13  eApplication                  74086 non-null  object
 14  Employment plan status        74086 non-null  int64
 15  Employability assessment      74086 non-null  int64
dtypes: int64(14), object(2)
memory usage: 9.0+ MB
```

Figure 1: General information on the dataset features

2.2 Data Understanding

The target variable, 'duration', is a numerical count variable. In order to gain a better understanding of the target variable, the probability distribution was plotted on a graph. Figure 2 shows the probability distribution of 'duration'.

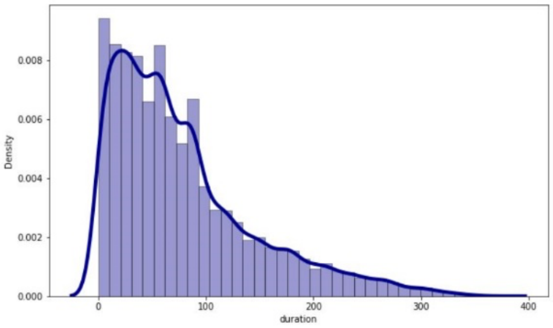


Figure 2: Probability distribution of the target variable

The information for the probability distribution of the target variable directly influences the predictive model selection. By looking at Figure 2, it can be assumed that that the target variable is following the Poisson distribution. We also plotted the distributions of the features. Figure 3 illustrates a grid of distributions of each feature of the dataset.

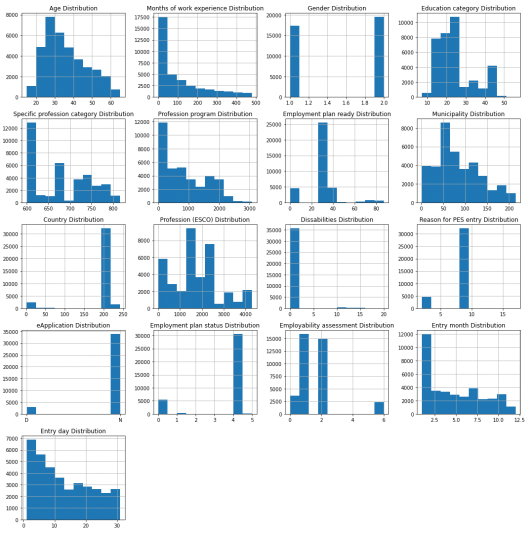


Figure 3: Grid of distributions of the dataset features

2.3 Data Preprocessing

It is estimated that in most data mining and knowledge discovery pipelines, 75 to 85% of the time is dedicated to preprocessing the data [5]. Cleaning and transforming samples are the cornerstone of a reliable and robust pattern recognition system. The first step of the data preprocessing part was data cleaning. The dataset included values for some of the attributes, which were an obvious result of a noise or a mistake. For example, some of the instances had negative values for the target variable, which is impossible because of the nature of that attribute, which is a count-based variable.

Most of the classical ML algorithms require the input data to be in numerical form. We used one-hot-encoding for the categorical features with at most 20 different categories. High-cardinality features were encoded using the Binary Encoding technique. Frequently used techniques like label-encoding do not work in high-cardinality because of the inclusion of artificial numerical relative distance between the instances or overfitting in the case of one-hot-encoding [6].

The ‘Entry Date’ feature was used to extract the day and month of entry separately. As those are cyclical features, we performed a transformation in order to better represent the cyclical phenomenon, for instance to avoid the artificial large difference between month 1 and month 12. The best way to handle this is to calculate the *sin()* and *cos()* component so that this cyclical feature is represented as (x, y) coordinates of a circle.

The normalisation of the attributes' values was applied to scale the attributes in a way that their mean value is zero, and their variance is retained with the use of their own standard deviation. It allows equality of opportunity for each attribute. By this, no attribute gives more value to itself regarding the range of values it has. Several normalisation techniques are commonly used, but the most popular one is the standard scaler, defined as:

$$z = \frac{x-\mu}{s} \tag{2.1}$$

where x is the actual value, μ is the mean, and s is the standard deviation.

All the calculations and transformations were performed in Python programming language, by making use of modules like pandas, NumPy and sci-kit learn.

3 METHODOLOGY

Since the target variable is numerical, the task should be treated as a regression problem. Regression analysis describes methods whose goal is to estimate the relationship between a dependent (target) variable and one or more independent variables. In formal terms, the goal is to specify the following general model

$$Y_i = f(X_i, \beta) + e_i \tag{3.1}$$

where i denotes the i^{th} observed input-output data set, the vector X represents the input (independent) variables, β is the set of model parameters, $f(\cdot)$ is the function, and e_i is the modelling error. The goal is to find the proper function f and its parameters β so the error term is as close to zero as possible.
In its simplest form, the function $f(\cdot)$ can represent a linear model. For example, the univariate linear model of (3.1) would be:

$$Y_i = \beta_0 + \beta_1 X_i + e_i \tag{3.2}$$

Generally, the function f can describe much more complex dynamics. The multivariate linear regression model is used as a base model and will be used to help with the assessment of the performance of other more specific and complex models simply by comparing them to the base model. The aim is to develop such models that will significantly outperform the base model. In order to construct a model that generalises well to the data, a decision tree is used as a base learning algorithm for the ensembles.

3.1 Ensemble Learning

The idea of ensemble learning is based on the theoretical foundations that the generalization ability of an ensemble is usually much stronger than the one of a single learner. Ensemble learning is mainly implemented as two subprocedures: training weak component learners and selectively combine the member learners into a stronger learner [7]. Two ensemble models based on different techniques were developed, Random Forest Regressor [8] and boosting algorithm - CatBoost Regressor.

Bagging is used to reduce the variance of a decision tree classifier. The objective is to create several subsets of data from the training sample chosen randomly with replacement. Each collection of subset data is used to train their corresponding decision trees. The result is the average of all the predictions from different trees, which is more robust than a single decision tree classifier.

Based on the shape of the probability distribution given in Figure 2, we assume that the target variable comes from Poisson distribution. Therefore, we design our model to maximise the log-likelihood for Poisson distribution [9]. The probability mass function of the Poisson distribution is given with the following expression:

$$P(k) = \frac{e^{-\lambda}(\lambda)^k}{k!} \tag{3.3}$$

where $P(k)$ is the probability of seeing k events during time unit given event rate λ. Let X, y be our dataset for the Poisson regression task. The log-likelihood function that needs to be maximised is:

$$\sum_{i=1}^{N} log\big(P_x(y)\big) = \sum_{i=1}^{N} log\left(\frac{e^{-\lambda(X_i)}\big(\lambda(X_i)\big)^{y_i}}{y_i!}\right) \tag{3.4}$$

After the expression is simplified, the final equation for the Poisson loss has the following form:

$$L_{poisson} = \sum_{i=1}^{N}\Big(\lambda(X_i) - y_i log\,\big(\lambda(X_i)\big)\Big) \tag{3.5}$$

CatBoost Regressor is optimised with regard to this objective function.

4 EVALUATION

The model performance on the test set is evaluated with Root Mean Squared Error (RMSE) as a metric. RMSE is frequently used in regression problems, and it is a measure of the difference between the values predicted by a model or an estimator and the actual values of the instances. RMSE is given with the following expression:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - pv_i)^2}{N}} \tag{3.6}$$

where y_i is the original value of the instance, and pv_i is the predicted value by the model. The hyper-parameters of the models were tuned using RandomizedSearchCV. This method optimises the hyper-parameters by cross-validated search over given parameter settings. A fixed number of parameter settings was sampled from the specified distributions.

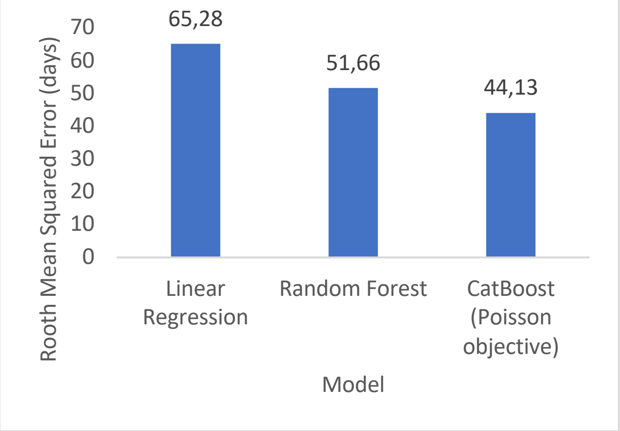


Figure 4: Comparison of the model performance

Figure 4 shows the diagram for comparison of the models' performances. The results show that both Random Forest and CatBoost significantly outperform the base linear regression model. Also, optimising the mean Poisson deviance as a loss function results in significant improvement in the performance of the boosting model. The final score that the CatBoost Regressor optimised with regards to mean Poisson deviance evaluated on RMSE is 44.13 days.

5 CONCLUSION

Achieving desirable results using machine learning models requires a significant amount of quality data and a deep understanding of the problem. Feature engineering is one of the key concepts here, which, if it is appropriately done, enables the generation of new features that give helpful, previously unknown insights about the data. The paper proposes an approach that emphasises the engineering of optimisation function concerning the probability distribution of the target variable, which results in developing a specific model for approaching the problem. Including the Poisson objective function in the boosting model resulted in significant improvement in its performance. There is still space for improvement in the results. Using modern end-to-end deep learning architectures have the potential to provide better results than the proposed models, which leaves space for future work on this topic. Having a tool that can roughly estimate the time a new client stays in the job-search process by having the standard data formation about himself is beneficial for the PES. The creation of decision-making tools for organisations dealing with employment services supports the process of reducing unemployment in the countries, which is a massive benefit for the global economy.

ACKNOWLEDGMENTS

First author acknowledges Ad Futura, Public Scholarship, Development, Disability and Maintenance Fund of the Republic of Slovenia. The second author acknowledges funding from the Slovenian Research Agency via program Complex Networks P1-0383. The last two authors acknowledge the funding received from the European Union’s Horizon 2020 research and innovation programme project HECAT under grant agreement No. 870702.

REFERENCES

[1] P. Boshkoski and B. Mileva - Boshkoska, "Report on commonly used algorithms and their performance," *Horizon 2020, Deliverable number: D3.1.,* 2020.

[2] J. Grundy, "Statistical profiling of the unemployed," *Studies in Political Economy,* 2015.

[3] T. Riipinen, "Risk profiling of long-term unemployment in finland," *Dialogue Con- ference Brussels.,* 2011.

[4] P. J. O'Connel, E. Kelly and J. Walsh, "National profiling of the unemployed in Ireland," *ESRI Research Series,* vol. 10, 2009.

[5] "HECAT - Disruptive Technologies Supporting Labour Market Decision Making," 2020. [Online]. Available: http://hecat.eu.

[6] F. Johannes, D. Gamberger and N. Lavrac, "Machine Learning and Data Mining," *Cognitive Technologies,* 2012.

[7] M. Brammer, Principles of Data Mining, 2007.

[8] F. Huang, G. Xie and R. Xiao, "Research on Ensemble Learning," *International Conference on Artificial Intelligence and Computational Intelligence,* 2009.

[9] A. Saha, S. Basu and A. Datta, "Random Forest for Dependent Data," *arXiv,* 2020.

[10] A. Zakariya Y, "Diagnostic in Poisson Regression Models," *Electronic Journal of Applied Statistical Analysis,* 2012.

Some Experimental Results in Evolutionary Multitasking

Andrejaana Andova
Jožef Stefan Institute and
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
andrejaana.andova@ijs.si

Bogdan Filipič
Jožef Stefan Institute and
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
bogdan.filipic@ijs.si

ABSTRACT

Transfer learning and multitask learning have shown that, in machine learning, common information in two problems can be used to build more effective models. Inspired by this finding, attempts in evolutionary computation have also been made to solve multiple optimization problems simultaneously. This new approach is called evolutionary multitasking (EMT).

In this work, we show how EMT extends ordinary evolutionary algorithms and present the results that we obtained in solving multiple optimization problems simultaneously. We also compare them with the results of algorithms that solve one optimization problem at a time. Finally, we provide visualizations and explanations of why and when EMT is beneficial.

KEYWORDS

evolutionary algorithms, numerical optimization, multifactorial optimization, evolutionary multitasking

1 INTRODUCTION

In optimization the task is to find one or more solutions that best solve a given problem. To determine which of the possible solutions gives the best result, we use the objective function. This can be the cost of fabrication, the efficiency of a process, the quality of a product, etc. The mathematical formulation of such problems is given as follows:

$$\begin{array}{rlr} \text{Minimize/Maximize} & f(x) & \\ \text{subject to} & g_j(x) \geq 0, & j = 1, 2, .., J; \\ & h_k(x) = 0, & k = 1, 2, .., K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, .., n. \end{array} \quad (1)$$

Here, a solution $x = [x_1, x_2, .., x_n]^{\mathrm{T}}$ is a vector of n decision variables. The objective $f(x)$ can be either maximized or minimized, but since many optimization algorithms are designed to solve minimization problems, we usually convert maximization objectives to minimization ones by multiplying the objective functions by -1. $h_k(x)$ are equality constraints, $g_j(x)$ inequality constraints, and $x_i^{(L)}$ and $x_i^{(U)}$ are boundary constraints [3]. In this paper, we consider problems that include only boundary constraints.

When the optimization problem can not be solved using mathematical methods, the usual alternative is to use randomized optimization algorithms such as evolutionary algorithms (EAs). These algorithms are characterized by a population of solutions that change with generations and to which techniques resembling natural selection and genetic variation are applied. These techniques ensure that the fittest individuals (solutions) from the population are passed to the next generation. The algorithm begins by initializing a population of solutions. Then, a selection operator is used to select the fittest individuals as parents. After that, a reproduction operator is utilized to create offspring from the parents. The next step is to select a subset of individuals from the combined set of parents and children and replace the old population with the selected subset. The new population is then used for the next generation. The cycle of selection, reproduction, and replacement is repeated until a stopping criterion is satisfied. The stopping criterion can be defined in various ways, for example, by the maximum number of generations.

Until recently, most EAs focused on solving only one optimization problem at a time. To exploit the parallelism of population-based search, Gupta et al. introduced a new category of optimization approach called multifactorial optimization or evolutionary multitasking (EMT) [8]. The goal of EMT is to develop EAs that are able to simultaneously solve multiple optimization problems without sacrificing the quality of the obtained solutions and the algorithm efficiency.

A practical motivation for the development of EMT algorithms is the rapidly growing cloud computing. In cloud computing, multiple users can simultaneously send optimization problems to the server. These problems may either have similar characteristics or they may belong to completely different domains. Previously, the servers solved these problems sequentially, but with the introduction of EMT, they can solve the problems in parallel.

After the introduction of EMT by Gupta et al., many other works followed that also introduced methodologies specialized in solving multiple optimization problems simultaneously [1, 4, 5, 6, 9, 10].

In this paper, we present our experimental results in solving multiple optimization problems simultaneously and discuss the results from the point of view of EMT performance. We do this by applying the EMT methodology as proposed by Gupta et al. to test optimization problems and analyzing the results.

The paper is further organized as follows. In Section 2, we introduce the basic concepts of EMT. In Section 3, we first present our results in EMT with visualizations that explain why and when EMT performs well, and then report the results in evolutionary many-task optimization. Finally, in Section 4, we give a conclusion and present the ideas for future work.

2 EVOLUTIONARY MULTITASKING

Evolutionary multitasking is characterized by the simultaneous existence of multiple decision spaces corresponding to different problems, which may or may not be independent, each with a unique decision space landscape. In order for EMT to have cross-domain optimization properties, Gupta et al. proposed to use a

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

uniform genetic code in which each decision variable is encoded with a random number from $[0, 1]$. Decoding such a representation in continuous problems is done by using the following equation for each decision variable:

$$u_i = u_i^{(L)} + (u_i^{(U)} - u_i^{(L)}) \cdot v_i, \tag{2}$$

where u_i is the decision variable in the original space, and v_i is the decision variable in the encoded space. The dimensionality of the solution vector is equal to $\max_j\{D_j\}$, where D_j represents the dimensionality of a single optimization problem. This type of encoding allows problems to share decision variables at the beginning of the genetic code, which contributes to the transfer of useful genetic material from one problem to another.

Since EMT attempts to solve multiple problems simultaneously using a single population, it is necessary to formulate a new technique for comparing population members. To this end, a set of additional properties is defined for each individual x_i in the population as follows.

- **Skill factor:** The skill factor τ_i of x_i is the one problem, among all problems in EMT, for which the individual is specialized. This skill factor can be assigned in a complex way by selecting the best individuals for each task or by randomly assigning each individual one task for which it is specialized. In our case, we will use the later, simpler method for assigning the skill factor.
- **Scalar fitness:** The scalar fitness is the fitness of an individual for the problem it is specialized.

To compare two solutions, we use the scalar fitness and the skill factor. The scalar fitness shows how good a solution is for a given problem, and the skill factor shows for which problem the solution performs best. A solution x_a is better than x_b if and only if both have the same skill factor and x_a has a higher scalar fitness than x_b. If the solutions have different skill factors, they are incomparable.

2.1 Assortative Mating

To produce offspring, the authors of EMT [8] used assortative mating as a reproduction mechanism. In assortative mating, two randomly selected parents can undergo crossover if they have the same skill factor. If, on the other hand, their skill factors differ, crossover occurs only with a given random mating probability *rmp*, otherwise, mutation takes place. A value of *rmp* close to 0 means that only culturally identical individuals are allowed to perform crossover, while a value close to 1 allows completely random mating.

2.2 Selective Imitation

Evaluating each individual for each problem is computationally expensive. For this reason, each child is evaluated only on one problem, which is the skill factor that one of its parents has. In this way, the total number of function evaluations is reduced, while the solution is still evaluated on the problem on which it most likely performs well. The procedure is called selective imitation.

2.3 Landscape Analysis

In multitask machine learning, it is well known that useful information cannot always be found for two problems. Therefore, to enable further success in the field of evolutionary multitasking, it is important to develop a meaningful theoretical explanation of when and why implicit genetic transfer can lead to improved performance. In particular, it is important to develop a measure of the inter-task complementarity used during the process of multitasking. To this end, a synergy metric that captures and quantifies how similar two problems are has been proposed [7]. The main idea behind the synergy metric is to use the dot product between the gradient of a given solution in one problem, and the vector pointing to the global optimum of another problem. If the dot product of a given solution is larger than 0, the solution of the first problem is pushing the candidate solution in the direction of the global optimum of the second problem. If the dot product is smaller than 0, the solution is pushed in the opposite direction.

3 EXPERIMENTS AND RESULTS

EMT is a novel concept in evolutionary optimization, and thus, a limited number of experiments were carried out so far. We present some experiments performed and results obtained using EMT in both multi- and many-task optimization.

3.1 Multitask Optimization

In the multitask optimization experiments, we took two frequently used optimization problems, i.e., 50-dimensional (50D) Sphere and Ackley. We solved them using EMT and a genetic algorithm (GA). To be able to compare the results, we used the same population size and the same number of function evaluations per problem. The *rmp* parameter in EMT was set to 0.3, and for GA we used the default parameter values as defined in pymoo [2]. We monitored the difference between EMT and GA over time. If the difference is positive, EMT performs better than GA, while if it is negative, GA performs better than EMT. Because the fitness values vary between different problems, we normalized the difference between EMT and GA in each problem by dividing the values with the highest absolute difference.

In the first experiment, the optima of the two problems were placed at the opposite ends of the search space. Because of this, the problems have very little common information, and the synergy function mostly takes negative values. This is visualized for a 2D Sphere function in Figure 1 and for a 2D Ackley function in Figure 2. The normalized difference between EMT and GA in optimizing 50D Sphere and Ackley functions is presented in Figure 3. From the results, we can see that GA performs better on these problems.

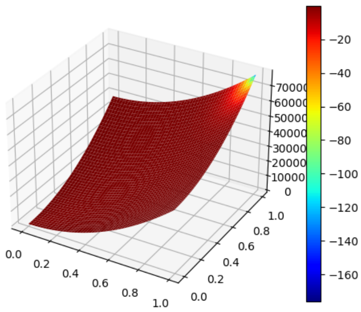


Figure 1: Synergy metric on the Sphere function solved together with the Ackley function when the optima are far away.

In Figure 4, we present the results from the second experiment where the optima of 50D Sphere and Ackley functions were placed closer together. Here, we can see that the optimization of the Sphere function does not show significant improvement

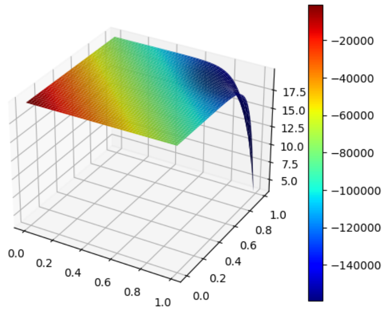


Figure 2: Synergy metric on the Ackley function solved together with the Sphere function when the optima are far away.

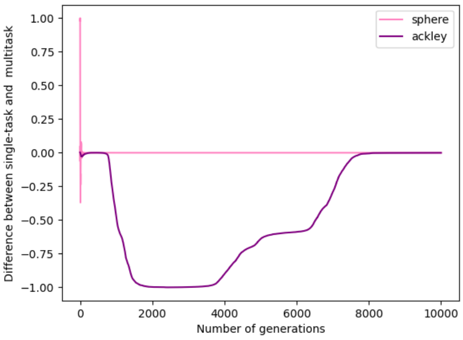


Figure 3: Normalized difference between multitask and single-task optimization on 50D Sphere and Ackley functions when the optima are far away.

when being performed together with the optimization of the Ackley function, but on the Ackley function EMT converges to the optimal solution much faster. An explanation for this is illustrated in 2D in Figures 5 and 6. Here we can see that the synergy in the Sphere space is mostly equal to 0, except for some small parts where it rises to +10 and falls to −10. Because both the positive and the negative parts of the synergy values of the Sphere problem are small, we can notice no difference in convergence on the Sphere problem.

In contrast, more than half of the space of the Ackley function has a positive synergy metric, indicating that this part of the space appoints the solutions in the right direction toward the global optimum. On the other hand, most of the decision space of the Ackley function has constant fitness values, which complicates the GA search for the global optimum. For this reason, the information transferred from the Sphere problem to the Ackley problem is useful, and thus we can see faster convergence when solving the two problems together using EMT.

3.2 Many-Task Optimization

When solving more than three tasks simultaneously, we are dealing with a many-task optimization. In Figure 7, we present the results obtained by randomly shifting (within a small, 10% range of the total space) the global optimum of both the Ackley and the Sphere function 25 times, resulting in 50 different 50D optimization problems. During the optimization process, we used the same algorithm parameter values for EMT and GA as reported in Section 3.1. In the results, we can notice similar patterns as when solving just two problems. This proves that increasing the number of problems we are trying to solve does not cause difficulties to EMT. If the problems are similar, we can solve many problems simultaneously without losing efficiency.

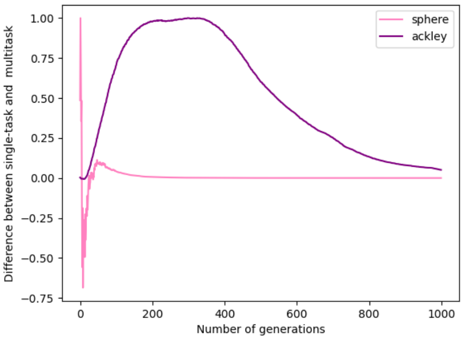


Figure 4: Normalized difference between multitask and single-task optimization on 50D Sphere and Ackley functions when the optima are close.

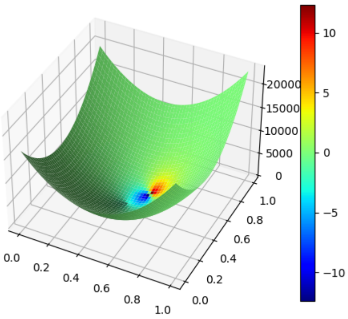


Figure 5: Synergy metric on the Sphere function solved together with the Ackley function when the optima are close.

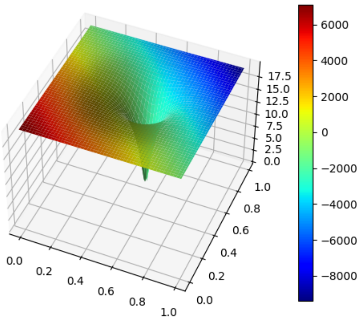


Figure 6: Synergy metric on the Ackley function solved together with the Sphere function when the optima are close.

Figure 8 shows the results obtained when solving six well-known optimization problems at the same time: Ackley, Sphere, Rastrigin, Rosenbrock, Schwefel, and Griewank, all 50D. From the results, we can notice that although the optimization procedure converges faster for most of the functions, for the Sphere and the Schwefel function the convergence speed of the optimization process drops. The same pattern can be noticed in Figure 9 where the optimum of each function is shifted 8 times, resulting in $6 * 8 = 48$ problems altogether.

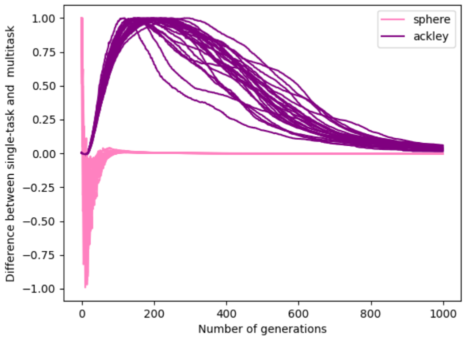


Figure 7: Normalized difference between multitask and single-task optimization on 50 problems originating from 50D Sphere and Ackley functions whose optima are shifted close to each other.

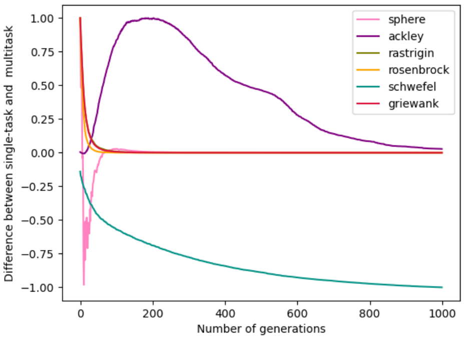


Figure 8: Normalized difference between multitask and single-task optimization on six well-known 50D optimization problems when the optima are shifted close to each other.

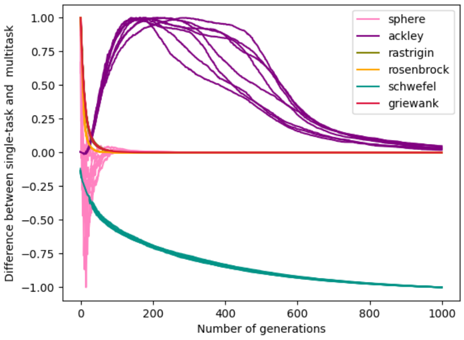


Figure 9: Normalized difference between multitask and single-task optimization on 48 problems originating from six well-known 50D optimization problems whose optima are shifted close to each other.

4 CONCLUSION AND FUTURE WORK

We presented our experimental results on solving multiple optimization problems simultaneously using a novel method called evolutionary multitasking. We were solving just two optimization problems, but also as many as 50 optimization problems at the same time. From the experimental results, we can conclude that there are some groups of problems for which EMT can improve the speed of convergence of the optimization process. However, if the problems are too different, the performance of the optimization drops. To explain why EMT works well on some problem pairs and why on some others it does not, we provided visualizations of the synergy metric.

We so far tested EMT on simple benchmark functions that are usually used for single-objective optimization. However, in future work, we plan to test it also on real-world scenarios with more complex functions and constraints. Furthermore, so far we have used the synergy metric to explain why some problems are solved Unfortunately, with this metric we can not strictly determine when solving two problems will be successful. Thus, one possible future direction is to develop machine learning methods that predict when multitasking a set of problems would be successful. This may be useful for cloud systems that could form several groups of similar problems and then solve them in a multitask manner.

5 ACKNOWLEDGMENTS

We acknowledge financial support from the Slovenian Research Agency (young researcher program and research core funding no. P2-0209).

REFERENCES

[1] Kavitesh Kumar Bali, Abhishek Gupta, Liang Feng, Yew Soon Ong, and Tan Puay Siew. 2017. Linearized domain adaptation in evolutionary multitasking. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1295–1302.

[2] Julian Blank and Kalyanmoy Deb. 2020. Pymoo: Multi-objective optimization in Python. *IEEE Access*, 8, 89497–89509.

[3] Kalyanmoy Deb. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester.

[4] Liang Feng, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Yew-Soon Ong, Kay-Chen Tan, and Alex Kai Qin. 2018. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 49, 9, 3457–3470.

[5] Maoguo Gong, Zedong Tang, Hao Li, and Jun Zhang. 2019. Evolutionary multitasking with dynamic resource allocating strategy. *IEEE Transactions on Evolutionary Computation*, 23, 5, 858–869.

[6] Abhishek Gupta, Jacek Mańdziuk, and Yew-Soon Ong. 2015. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems*, 1, 1-4, 83–95.

[7] Abhishek Gupta, Yew-Soon Ong, Bingshui Da, Liang Feng, and Stephanus Daniel Handoko. 2016. Landscape synergy in evolutionary multitasking. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 3076–3083.

[8] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. 2015. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20, 3, 343–357.

[9] Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. 2016. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 47, 7, 1652–1665.

[10] Yu-Wei Wen and Chuan-Kang Ting. 2017. Parting ways and reallocating resources in evolutionary multitasking. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2404–2411.

Intent Recognition and Drinking Detection For Assisting Kitchen-based Activities

Carlo M. De Masi
carlo.maria.demasi@ijs.si
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia

Simon Stankoski
simon.stankoski@ijs.si
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia

Vincent Cergolj
vc2756@student.uni-lj.si
Univerza v Ljubljani, Fakulteta za elektrotehniko
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia

Mitja Luštrek
mitja.lustrek@ijs.si
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

We combine different computer-vision (pose estimation, object detection, image classification) and wearable based activity recognition methods to analyze the user's behaviour, and produce a series of context-based detections (detect locations, recognize activities) in order to provide real-time assistance to people with mild cognitive impairment (MCI) in the accomplishment of every day, kitchen-related activities.

KEYWORDS

computer vision, activity recognition, object detection, pose estimation

1 INTRODUCTION

Smart home technologies have been extensively adopted for measuring and decreasing the impact of Mild Cognitive Impairment (MCI) on everyday life [9]. In the scope of the CoachMyLife (CML) project we have been developing a system employing different machine learning techniques with the aim of assisting persons affected by MCI in performing activities in their apartments, with a particular focus on tasks related to the kitchen.

In a previous work, we presented one of the first components of this system, i.e. a computer vision pipeline which allows to detect the activity of drinking, by analyzing the video collected by an RGB camera through a 3D Convolutional Neural Network (3D-CNN) [12].

In the present paper, we present our work on extending said pipeline, by discussing (i) a drinking-detection algorithm based on motion data from a wristband, which can be used to further validate the one based on computer vision, and to replace it in situations where the activity is not performed in front of the camera; (ii) a method based on pose detection to identify interactions of the user with their environment, in order to perform intent recognition, and (iii) a possible new implementation of our previous computer-vision pipeline for drinking detection that can be deployed on edge devices.

This paper is organized as follows. Section 2 discusses the related work. Section 3 presents the system architectures. Section 4 describes the recognition modules of the system. Section 5 shows the results of the recognition modules. Finally, Section 6 concludes the paper.

2 RELATED WORK

2.1 Drinking Detection From Wearables

Recent advances in the accuracy and accessibility of wearable sensing technology (e.g., commercial inertial sensors, fitness bands, and smartwatches) has allowed researchers and practitioners to utilize different types of wearable sensors to assess fluid intake in both laboratory and free-living conditions.

The necessity for fluid intake monitoring emerges as a result of people's lack of awareness of their hydration levels. Dehydration can lead to many severe health problems like organ and cognitive impairments. Therefore, a system that can continuously track the fluid intake and provide feedback to the user if useful.

In [1], the authors explored the possibility of recognizing drinking moments from wrist-mounted inertial sensors. They used adaptive segmentation to overcome the problem with variable length of the drinking gestures. They used random forest algorithm, trained with 45 features, and obtained an average precision of 90.3% and an average recall of 91.0%. In [5], the authors employed a two-step detection procedure, enabling them to detect drinking moments and estimate the fluid intake. They extracted 28 statistical features, from which only six were selected using backward feature selection. Finally, they trained a Conditional Random Field model, resulting in a precision of 81.7% and recall of 77.5%. In [4], the authors used a machine-learning based model to detect hand-to-mouth gestures. Similarly as the previous methods, they extracted 10 time-domain features and trained a random forest classifier. They validated their method in a free-living scenario and obtained precision of 84% and recall of 85%. Although remarkable results were achieved, the evaluation of the studies is limited and it is not showing the real-life performance.

2.2 Activity Recognition From Videos

In recent years, the problem of computer-based Human Activity Recognition (HAR) of daily living has been tackled by different computer-vision methods.

HAR can be performed directly on RGB images and videos by analyzing: (i) the spatial features in each frame, thus obtaining

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

predictions for each frame that can then be extended to the whole video by pooling or by a recurrent-based neural networks [2], (ii) the temporal features related to motions and variations between frames [6], or (iii) some combination of the two [10].

The most recent approaches aimed at simultaneous evaluation of both spatial and temporal features involve the usage of 3D-CNN, i.e., convolutional models characterized by an additional third temporal dimension [12].

An alternative approach, not involving the direct analysis of the whole frames, consists in exploiting the information provided by human pose estimation, so that body keypoints coordinates, reconstructed in a 2D or 3D space, can be fed to deep-learning models to provide predictions [3].

3 ADOPTED HARDWARE

3.1 Wristband

The drinking-detection procedure is implemented on a wristband which is equipped with a nRF52840 System On Chip (SoC) module. The SoC offers a large amount of Flash and RAM, 1MB and 256 kB, respectively. Additionally, it has protocol support for Bluetooth Low Energy (BLE). The architecture of nRF52840 is based on 32-bit ARM® Cortex™-M4 CPU with floating point unit running at 64 MHz. The wristbands power supply source is a battery with a capacity of 500 mAh. The measurements of accelerations and angular velocities are performed by the system-in-package LSM6DSL, manufactured by STM. It is equipped with a 3D digital accelerometer and a 3D digital gyroscope based on MEMS technology that operates at 0.65 mA in high-performance mode and allows low power consumption with constant operation. The most prominent feature of the Inertial Measurement Unit (IMU) is a 4 kB FIFO (First In First Out) buffer, which stores the data of the accelerometer and gyroscope. This allows for very low power operation, as the SoC wakes up only when triggered by an "FIFO full" interrupt event.

3.2 Local Deployment of The Computer Vision System

The computer vision pipeline for drinking detection we previously developed for the project worked by retrieving the video stream collected by an IP camera in the user's apartment, and analyzing it on a remote server. This approach, however, presented issues related to the remote access to the camera, which can sometimes be blocked by the router's firewall functionalities, and raised safety and privacy concerns with the users.

For these reasons, we have been working on deploying the CML system on a local device. After some unsuccessful attempts to implement the system on Android devices by using frameworks such as Apache TVM [1] or Deep Java Library (DJL)[2], we opted for deployment on a Jetson NANO device[3].

Direct deployment of our system on the device was possible, although not immediate, but the resulting performance was suboptimal in terms of the FPS reached by the various detection algorithms (≈2 FPS for the object detection). To overcome this, we optimized said algorithms by TensorRT, a library built on NVIDIA's CUDA library for parallel programming, thus improving inference performance for deep learning models (≈22 FPS for the object detection).

4 INTENT RECOGNITION

One of the main goals of the CML project is to provide users with real-time, context-based notifications to assist them in performing activities.

This is achieved in two steps. First, by combining computer-vision and the wearable device, the system detects real-time events, such as the position of the user, their interaction with the environment, the displacement of a mug the user is expected to drink from, the opening/closing of cabinet and fridge door, drinking and eating.

Then, these events are passed to the intent recognition module, which uses them to predict which activity the user is performing, and provide assistance if needed.

We adopted a Single Shot MultiBox Detector (SSD) [8] model, pre-trained on the 80 classes of the COCO dataset [7] for the detection of the user, and fine-tuned on a custom dataset we collected to locate the position of the mug. Pose estimation, which is used to track the movement of the user's hands and detect interactions with domestic appliances, is achieved by a SimpleNet model with a ResNet backbone [13].

4.1 Regions of Interest

During the initial setup, the user is asked to identify some regions of interest (ROIs) in the camera image, which can be either single or double-zoned.

In the first case, the ROI is "activated" when the user's feet are within the selected region (Fig. 1a), whereas double-zone ROIs are used to detect if the user is in the desired area and/or if their hands are in the selected upper area (Fig. 1b).

4.2 Intent Recognition

The events detected by the computer vision pipeline are passed to the intent recognition module, which predicts the activity the user is currently engaged on.

Currently, this prediction is based on a set of pre-determined rules. A number of possible activities is manually inserted, each formed by different steps, corresponding to possible events that can be detected by the computer vision system (Fig. 2a). Different activities can share one or more steps, and as the system detects the completion of the various steps, the list of possible ongoing activities gets reduced (Fig. 2b, 2c), until only one activity is identified and followed until its completion (Fig. 2d).

If too long of a time interval passes between the completion of two steps, the activity is classified as "interrupted", and the system can show a notification to the user, asking if they require assistance.

4.3 Drinking Detection From Computer Vision on the Jetson NANO

The model we previously adopted to perform activity recognition from videos is particularly computationally expensive and so, although it proved to be very effective in the detection of drinking events, it was not possible to implement it on the Jetson NANO.

For this reason, we are currently collecting a dataset of short video clips, passing them through a pose estimation model, in order to obtain the 2D position of 18 body parts across a time series of frames, with an associated class label for the frame series. Then, this will be analyzed through an LSTM-based model to perform HAR.

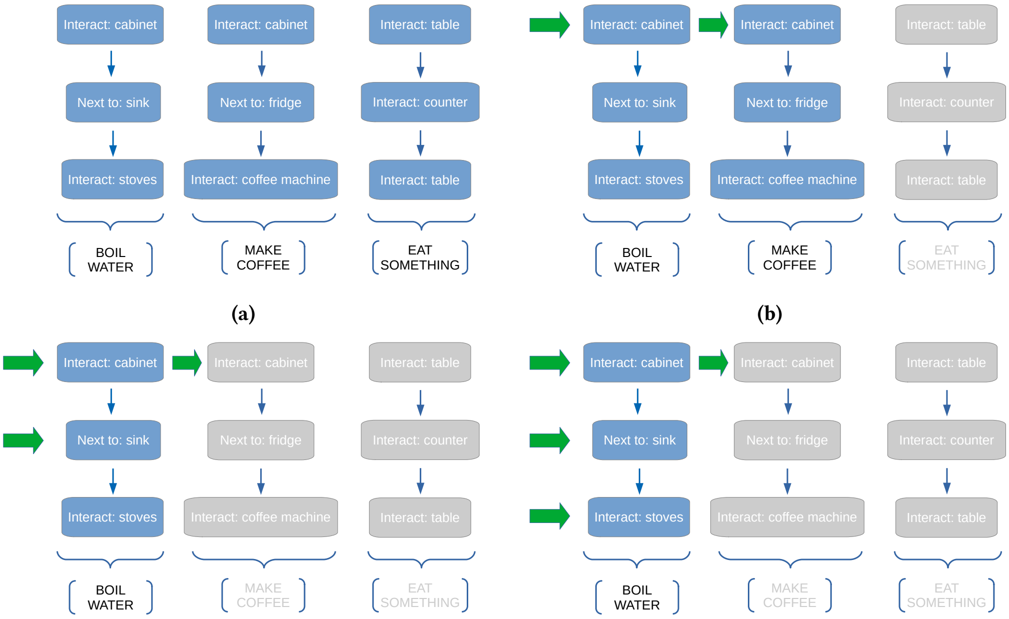
[1] https://tvm.apache.org/

[2] https://djl.ai/

[3] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/

(a) (b)

Figure 1: Triggers based on user's location and their interaction with the environment.



(c) (d)

Figure 2: As the computer vision system detects the completion of various steps, the list of possible ongoing activities gets reduced, until one of them is completed or interrupted.

4.4 Drinking Detection Using a Wearable device

Due to the desired minimum power consumption, the drinking detection was implemented directly on the wristband. This is preferable as it eliminates the need to transfer all the raw sensor data to a smartphone or some sort of central device. Raw sensor data transmission is clearly undesirable due to high power consumption and it is not possible if the central device is not nearby.

The first step of drinking detection using the wristband is to enable the IMU in activity/inactivity recognition mode. This allows the IMU to be in a low power state for the most part of the day.

When activity is recognized the IMU enables absolute wrist detection (AWT) which checks if the angle between the horizontal plane and the Y axis of the IMU is larger than 30 degrees. If the condition is met the IMU is enabled in batching mode, storing accelerometer and gyroscope data in the FIFO buffer. Every time the FIFO buffer is full, data is transferred to the SoC, where we directly start the machine learning pipeline. This procedure is repeated for three batches of IMU readings. If all three predictions from the machine learning model are non drinking, we disable the gyroscope, we stop the machine learning procedure and we wait for the next AWT event. Otherwise, if at least one prediction is positive, the machine-learning procedure continues to work for another three new batches of data.

The machine-learning method for detection of drinking gestures is based on time- and frequency-domain features. The raw data is segmented into 5-second windows and 216 features are extracted in total. We used a relatively simple approach due to the memory limitation of the wristband. The deployed model was trained using the drinking dataset described in Section 4.4.1 and additional non-drinking data collected in real-life scenario [11].

4.4.1 Drinking Dataset.

For the aim of this study, we recruited 19 subjects (11 males and 8 females). Each subject was equipped with the wristband described in Section 3.1. We developed a custom application that ran on the wristband and collected three-axis accelerometer and three-axis gyroscope data at a sampling frequency of 50 Hz. The dataset [4] is publicly available and we hope that it will serve researchers in future studies.

We developed a general procedure for the participants to follow during the data collection process. The ground truth was registered manually by participants pressing a button on the wristband before performing the gesture and after finishing the

[4] https://github.com/simon2706/DrinkingDetectionIJS

gesture. The data collection procedure included drinking from six different container types—namely, bottle, coffee cup, coffee mug, glass, shot glass and wine glass.

For each participant we collected 36 drinking episodes (3 fluid level x 6 containers x 2 positions). The idea of the different fluid level was to obtain drinking episodes with a short, medium and long duration. We also considered different body positions. The participants first performed the drinking gestures while being seated and afterwards they repeated the same gestures while standing.

5 RESULTS AND DISCUSSION

5.1 Intent Recognition and Local Implementation of Drinking Detection

A pilot phase will begin shortly, during which the intent recognition module will be evaluated.

Regarding the new model for drinking detection, a preliminary test of our new approach, ran on a subset of the Berkeley Multimodal Human Action Database (MHAD) dataset [5], reached an accuracy of over 90%, and we'll extend the analysis to our case once the dataset collection will be over.

5.2 Wearable Sensing Results

For evaluation, the leave-one-subject-out (LOSO) cross-validation technique was used. In other words, the models were trained on the whole dataset except for one subject on which we later tested the performance.

For the drinking detection model, we considered several classifiers including logistic regression (LR), linear discriminant analysis (LDA), k-nearest neighbors (KNN), naive Bayes (NB) and XGBoost.

The obtained results are shown in Table 1. It can be clearly seen that XGBoost outperforms all other classifiers. However, due to the technical limitations described in Section 3.1 the trained model is unable to fit below 100 KB. The size of the LR model is only 2 KB, which is optimal for our device. Furthermore, the results obtained with LR are only 0.03 lower compared to those from XGBoost. Therefore, we deployed the model trained with the LR classifier.

6 CONCLUSIONS

We presented our work on drinking detection using wearables and intent recognition/drinking detection using computer vision.

A pilot phase, beginning in October 2021, will provide thorough testing of the functionalities described in the paper. Nonetheless, the results obtained from the internal testing for each module of the system show promising results for both drinking (with both wearables and computer vision) and intent recognition.

[5] http://tele-immersion.citris-uc.org/berkeley_mhad

Table 1: Comparison of different classifiers for detection of drinking activity.

Method	Precision	Recall	F1 score
Logistic regression	0.87	0.77	0.81
Linear discriminant analysis	0.54	0.69	0.55
K-nearest neighbors	0.84	0.69	0.75
Naive Bayes	0.68	0.85	0.74
XGBoost	0.89	0.81	0.84

REFERENCES

[1] Keum San Chun, Ashley B Sanders, Rebecca Adaimi, Necole Streeper, David E Conroy, and Edison Thomaz. 2019. Towards a generalizable method for detecting fluid intake with wrist-mounted sensors and adaptive segmentation. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 80–85.

[2] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, et al. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634.

[3] Giovanni Ercolano and Silvia Rossi. 2021. Combining cnn and lstm for activity of daily living recognition with a 3d matrix skeleton representation. *Intelligent Service Robotics*, 14, 2, 175–185.

[4] Diana Gomes and Inês Sousa. 2019. Real-time drink trigger detection in free-living conditions using inertial sensors. *Sensors*, 19, 9, 2145.

[5] Takashi Hamatani, Moustafa Elhamshary, Akira Uchiyama, and Teruo Higashino. 2018. Fluidmeter: gauging the human daily fluid intake using smartwatches. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2, 3, 1–25.

[6] Ammar Ladjailia, Imed Bouchrika, Hayet Farida Merouani, Nouzha Harrati, and Zohra Mahfouf. 2020. Human activity recognition via optical flow. *Neural Computing and Applications*, 32, 21, 16387–16400.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. 2014. Microsoft coco: common objects in context. (2014). arXiv: 1405.0312 `[cs.CV]`.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. 2016. Ssd: single shot multibox detector. *Lecture Notes in Computer Science*, 21–37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2. http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[9] Maxime Lussier, Stéphane Adam, Belkacem Chikhaoui, Charles Consel, Mathieu Gagnon, Brigitte Gilbert, Sylvain Giroux, Manon Guay, Carol Hudon, Hélène Imbeault, et al. 2019. Smart home technology: a new approach for performance measurements of activities of daily living and prediction of mild cognitive impairment in older adults. *Journal of Alzheimer's Disease*, 68, 1, 85–96.

[10] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576.

[11] Simon Stankoski, Marko Jordan, Hristijan Gjoreski, and Mitja Luštrek. 2021. Smartwatch-based eating detection: data selection for machine learning from imbalanced data with imperfect labels. *Sensors*, 21, 5. ISSN: 1424-8220. DOI: 10.3390/s21051902. https://www.mdpi.com/1424-8220/21/5/1902.

[12] Gül Varol, Ivan Laptev, and Cordelia Schmid. 2017. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40, 6, 1510–1517.

[13] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, 466–481.

Anomaly Detection in Magnetic Resonance-based Electrical Properties Tomography of in silico Brains

Ožbej Golob
University of Ljubljana
Faculty of Computer and
Information Science
Ljubljana, Slovenia
ozbej.golob@gmail.com

Alessandro Arduino
Istituto Nazionale di Ricerca
Metrologica
Torino, Italy
a.arduino@inrim.it

Oriano Bottauscio
Istituto Nazionale di Ricerca
Metrologica
Torino, Italy
o.bottauscio@inrim.it

Luca Zilberti
Istituto Nazionale di Ricerca
Metrologica
Torino, Italy
l.zilberti@inrim.it

Aleksander Sadikov
University of Ljubljana
Faculty of Computer and
Information Science
Ljubljana, Slovenia
aleksander.sadikov@fri.uni-lj.si

ABSTRACT

Magnetic resonance-based electrical properties tomography (EPT) is one of the novel *quantitative* magnetic resonance imaging techniques being tested for use in clinical practice. This paper presents preliminary research and results of automated detection of anomalies from EPT images. We used in silico data based on anatomical human brains in this experiments and developed two algorithms for anomaly detection. The first algorithm employs a standard approach with edge detection and segmentation while the second algorithm exploits the quantitative nature of EPT and works directly with the measured electrical properties (electrical conductivity and permittivity). The two algorithms were compared on – as of yet – noiseless data. The algorithm using the standard approach was able to quite reliably detect anomalies roughly the size of a cube with a 14 mm edge while the EPT-based algorithm was able to detect anomalies roughly the size of a cube with a 12 mm long edge.

KEYWORDS

electrical properties tomography (EPT), magnetic resonance imaging (MRI), automatic anomaly detection, artificial intelligence

1 INTRODUCTION

The frequency-dependent electrical properties (EPs), including electrical conductivity and permittivity, of biological tissues provide important diagnostic information, e.g. for tumour characterisation [9]. EPs can potentially be used as biomarkers of the healthiness of various tissues. Previous studies, not based on magnetic resonance imaging (MRI), have shown that various diseases cause changes of EPs in the tissue [3].

Electrical properties tomography (EPT) is used for quantitative reconstruction of EPs distribution at radiofrequency (RF) with spatial resolution of a few millimetres. EPT requires no electrode mounting and, during MRI scanning, no external energy is introduced into the body other than the B_1 fields. Applied B_1 fields can easily penetrate into most biological tissues, making EPT suitable for imaging of the whole body. The MRI scans for EPT are performed using a standard MRI scanner, and its spacial resolution is determined by MRI images and quality of used B_1-mapping technique [9].

The objective of this research was to develop and evaluate algorithms to automatically detect anomalies of different sizes in the EPT images. The data consisted of in silico simulated brain scans of phantoms that either contained an anomaly or not. The evaluation was aimed towards answering whether an anomaly can be detected or not, and how large an anomaly can be (reasonably) reliably detected. This represents an initial step towards the potential clinical use of EPT.

2 METHODS

2.1 Data Acquisition

The MRI acquisition of the EPT inputs has been simulated in a noiseless case. Thus, the result of the electromagnetic simulation at RF has been directly converted in the acquired data, with no further post-processing. Precisely, the B_1 field generated by a current-driven 16-leg birdcage body-coil (radius 35, height 45) operated both in transmission and in reception with a polarisation switch has been computed in presence of anatomical human heads with a homemade FEM–BEM code [2]. The simulations have been conducted at 64 (i.e. the Larmor frequency of a 1.5 scanner).

The acquisitions of 19 human head models from the XCAT library [6] have been simulated. The considered population is statistically representative of different genders and ages. For each head model, 10 different variants are considered:

(1) Two physiological variants with the original distribution of the biological tissues. In one case, the nominal electrical conductivity provided by the IT'IS Foundation database [5] is assigned to each tissue. In the other case, the electrical conductivity of white and grey matter is sampled from a uniform distribution that admits a variation up to 10 with respect to the nominal value. This will be referred to as the *physiological variability* of the electrical conductivity.
(2) Eight pathological variants, in which a spherical pathological inclusion is inserted in the white matter tissue. The radius of the inclusion ranges from 5 to 45 and its electrical conductivity is set equal to that of the white

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

matter increased by a factor uniformly sampled from 10 to 50 of the nominal value, because previous experimental results have shown that pathological tissues have higher EP values than healthy tissue [7, 8]. The location of the inclusion within the head is selected with a random procedure and only its intersection with the white matter tissue is kept in the final model (see Fig. 1 panels a and d). All the pathological variants take into account the physiological variability in the determination of the white and grey matter electrical conductivity.

2.2 Reconstruction Techniques

In order to retrieve the distribution of the electrical conductivity, the phase-based implementations of Helmholtz-EPT (H-EPT) and convection-reaction–EPT (CR-EPT) provided by the open-source library EPTlib [1] have been used. For each head model, the distribution of the *transceive* phase [3] (input of phase-based EPT) is obtained by linearly combining the phases of the rotating components of B_1 simulated both in transmission and in reception [1].

Since noiseless inputs are considered, the smallest filter has been used both in H-EPT and in CR-EPT. Moreover, CR-EPT has been applied for a volume tomography, with an electrical conductivity of 0.1 forced at the boundaries and an artificial diffusion coefficient equal to 10^{-4} .

Currently, the proposed anomaly detection algorithms have been tested only on the H-EPT results.

2.3 Anomaly Detection

We developed two anomaly detection algorithms: (i) a more classical approach for anomaly detection in MR images and (ii) an EPT-based approach working with direct quantitative properties estimated by the MRI-based EPT.

2.3.1 *Classical Approach.*

The classical approach uses standard techniques used for anomaly detection in MR images. This approach could be applied (also) on standard MR images as it is independent of the MRI technique. The algorithm uses noiseless EPT images, produced with Helmholtz reconstruction technique, as input data.

The algorithm receives previously segmented (this segmentation was not of interest in this research) white matter from the EPT image and detects the edges in it. The edges are detected using a simple gradient edge detection technique. The gradient is calculated for each voxel based on the directional change of electrical conductivity of neighbouring voxels. The edges are represented as borders between white matter and other brain tissues as well as borders between white matter and anomalies. Edge voxels are ignored in order to avoid H-EPT reconstruction errors, which occur at borders between tissues [4].

The algorithm then calculates median electrical conductivity of all regions as separated by the detected edges. Figure 1 shows median electrical conductivity distribution by regions in a sample image.

The k-means algorithm is then employed for the classification of regions into healthy and anomaly-containing ones. The algorithm classifies an MR image based on median electrical conductivity of each region. The anomaly location is associated with the regions detected as containing the anomaly.

2.3.2 *EPT Approach.*

EPT differs from standard MRI techniques by representing EPs as quantitative values. EPs are a reliable biomarker of healthy brain. Mandija et al. [4] presented mean electrical conductivity and standard deviation of white and grey matter as a reliable measure of whether the brain contains pathological tissue.

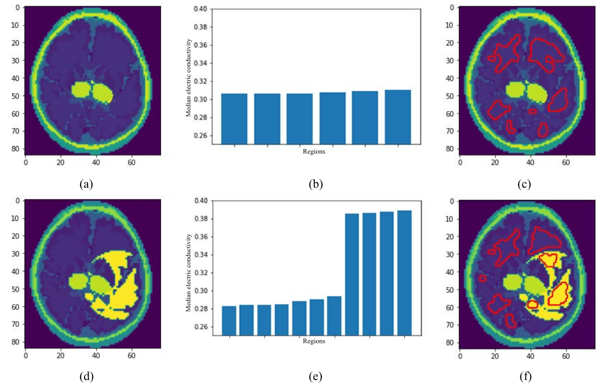


Figure 1: Median electrical conductivity distribution by regions. (a) Segmented healthy MRI image. (b) Median electrical conductivity distribution. (c) Detected regions (bordered red). (d) Segmented pathological MRI image (anomaly is yellow). (e) Median electrical conductivity distribution. (f) Detected regions (bordered red). Please note that not all of the regions are visible as only a 2D slice is shown while the data is 3D.

In input data for our experiments, electrical conductivity is distributed from 90% to 110% of nominal value for white matter, and from 110% to 150% for anomalies. However, it must be noted that these are the values used for setting up the phantoms, and that these values are then only approximated when EPT reconstruction is performed. These reconstructed properties have been used as input for anomaly detection. The algorithm detects anomalies based on the difference between white matter and anomalies. The algorithm uses noiseless EPT images, produced with H-EPT, as input data.

The algorithm, as the classical one, receives as input previously segmented white matter from the whole EPT image. It then detects all voxels that have electrical conductivity between 110% and 150% of median electrical conductivity of white matter and marks them as a potential anomaly. These voxels, marked as potentially being an anomaly, are then grouped into regions based by their location. The algorithm ignores all smaller regions (below a set size threshold) that likely represent noise and reconstruction errors. All the remaining regions are classified as the anomaly.

3 RESULTS

Figure 2 shows the predictions of whether an image contains an anomaly or not for both algorithms – classical on the left (a) and EPT approach on the right (b). Each EPT image corresponds to one bar on the chart and they are arranged with the increasing size of the anomaly; the size of the bar represents the size of the anomaly in voxels. The bars are cut off at 2,000 voxels for easier viewing. Only images actually containing the anomaly are shown; for the others the false positives (FP) rate describes the performance of the two algorithms. The green colour represents correct predictions and the red colour the incorrect ones. The yellow colour means that the algorithm correctly predicted the

Table 1: Classification evaluation of the classical approach.

Measure	Training data	Test data
Precision	0.975	1.000
Recall	0.750	0.708
F1 score	0.848	0.829
Accuracy	0.785	0.767

Table 2: Localisation evaluation of the classical approach.

Measure	Training data	Test data
IoU	0.197 ± 0.116	0.244 ± 0.110
Precision	0.932 ± 0.202	0.988 ± 0.050
Recall	0.204 ± 0.123	0.245 ± 0.110
F1 score	0.313 ± 0.163	0.379 ± 0.143

presence of the anomaly, but for the wrong reasons (hence Intersection over Union (IoU) is zero) – these cannot be counted as correct performance. Some misclassifications are labeled with the most likely cause: either that the anomaly is scattered in several smaller regions (each below the detection threshold size) or, in case of the EPT approach, that the anomaly is too close to the top border and is ”overshadowed” by the cranium. For the unlabelled misclassifications the most likely reason is the small size of the anomaly.

Figure 2 captures rather well the minimal anomaly size where each algorithm starts performing quite reliably. The classical approach detects anomalies larger than 350 voxels and the EPT approach detects anomalies larger than 170 voxels. Since each voxel represents a cube with a 2 mm edge, these volumes translate roughly to a cube with the edge of 14 mm for the classical approach and a cube with the edge of slightly less than 12 mm for the EPT approach.

Tables 1-4 further clarify the results. The images were split into a training set, used to optimise several internal parameters and a test set for independent evaluation. Internal parameters of the classical approach specify: (i) minimum gradient value for a voxel to be recognized as an edge; (ii) electrical conductivity difference between anomaly and healthy tissue; (iii) minimum region size. Internal parameters of the EPT approach specify: (i) how many initial slices of white matter are ignored (to avoid reconstruction errors); (ii) minimum region size. The split, while random in nature, was made based on individual phantom heads – the same head with different anomalies simulated could not be both in the test and training set. The training set consisted of 130 images (including 26 not containing an anomaly), and the test set consisted of 60 images (including 12 not containing an anomaly).

Table 1 shows the results of classification evaluation of the classical approach and Table 2 shows the results of localisation evaluation using the classical approach. The localisation results are reported as mean ± standard deviation of electrical conductivity. The values of IoU and F1 score for localisation are lower as a result of ignoring anomaly edge voxels. Anomaly edge voxels are ignored because of H-EPT reconstruction errors. This is not an issue for anomaly detection as values of precision are still high. Values of IoU and F1 score of localisation will be improved by acknowledging edges of anomaly after it is already detected.

Table 3: Classification evaluation of the EPT approach.

Measure	Training data	Test data
Precision	0.976	0.971
Recall	0.769	0.708
F1 score	0.860	0.819
Accuracy	0.800	0.750

Table 4: Localisation evaluation of the EPT approach.

Measure	Training data	Test data
IoU	0.381 ± 0.140	0.435 ± 0.125
Precision	0.874 ± 0.208	0.900 ± 0.177
Recall	0.396 ± 0.142	0.450 ± 0.126
F1 score	0.535 ± 0.166	0.594 ± 0.142

Analogously, Table 3 shows the results of classification evaluation of the EPT approach and Table 4 shows the results of localisation evaluation of the EPT approach. Again, IoU and F1 score values are reduced as the result of ignoring anomaly edge voxels.

An example of anomaly localisation is shown in Figure 3. As shown in the image, the EPT approach is generally better at anomaly localisation than the classical approach.

4 DISCUSSION AND CONCLUSIONS

The results indicate potential for future use of the EPT technique for the anomaly detection in clinical practice. The results in terms of the anomaly size are on par with what a trained radiologist is able to detect manually.

EPT, being a quantitative technique, offers the advantage of comparability of the images (e.g. in longitudinal monitoring of the patient) compared to the standard qualitative MRI. Furthermore, the direct EPT approach performed better than the classical one via edge detection. It is also less complex and this can often be a bonus in practical applications.

However, this is a pilot study and further research is required to put these approaches into actual practice. The biggest limitation of the presented study and results is that the images, while being an actual EPT reconstruction, were deliberately noiseless. With the introduction of noise the data would very much resemble the actual in vivo cases, however the obtained results will likely be worse. A lot of further work, mostly on noise reduction and detection in presence of noise is likely still required.

Moreover, currently only the data captured using H-EPT is used. This technique causes (large) reconstruction errors which occur at the borders between tissues. The results could potentially be improved by combining H-EPT and CR-EPT [1], as the latter technique does not cause reconstruction errors at borders between tissues.

The anomaly localisation could also be improved by not ignoring edges. The edges would still be removed when anomalies are detected, however, once an anomaly is detected, the edges around the anomaly could be classified as anomaly, thus improving the IoU and the F1 score.

In addition to the mean value of electrical conductivity, the standard deviation of the electrical conductivity could also be taken into account when detecting edges and anomalies.

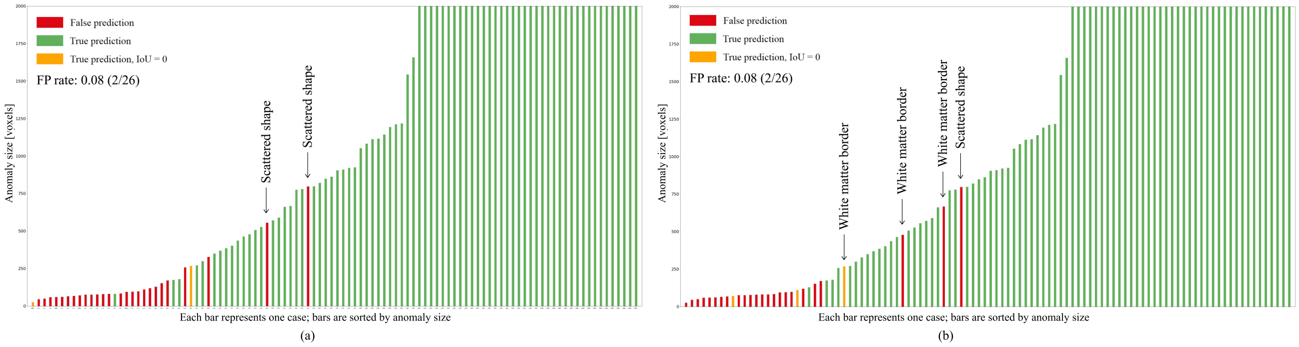


Figure 2: Predictions of anomaly detection algorithms. (a) Classical approach. (b) EPT approach.

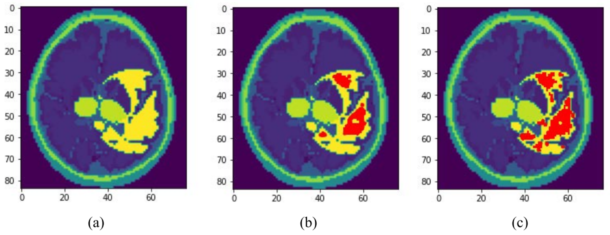


Figure 3: Anomaly localization. (a) Segmented pathological MRI image. (b) Localization of classical approach (detected anomaly is red). (c) Localization of EPT approach (detected anomaly is red).

Finally, once results achieved on EPT images of phantom brain are satisfactory, implemented approaches could be tested on in vivo data.

ACKNOWLEDGMENTS

The results presented here have been developed in the framework of the EMPIR Project 18HLT05 QUIERO. This project has received funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme.

REFERENCES

[1] A. Arduino. 2021. EPTlib: an open-source extensible collection of electric properties tomography techniques. *Applied Science*, 11, 7, 3237.

[2] O. Bottauscio, M. Chiampi, and L. Zilberti. 2014. Massively parallelized boundary element simulation of voxel-based human models exposed to MRI fields. *IEEE Transactions on Magnetics*, 50, 2, 7025504.

[3] Jiaen Liu, Yicun Wang, Ulrich Katscher, and Bin He. 2017. Electrical properties tomography based on B_1 maps in MRI: principles, applications, and challenges. *IEEE Transactions on Biomedical Engineering*, 64, 11, 2515–2530. DOI: 10.1109/TBME.2017.2725140.

[4] Stefano Mandija, Petar I. Petrov, Jord J. T. Vink, Sebastian F. W. Neggers, and Cornelis A. T. van den Berg. 2021. Brain tissue conductivity measurements with MR-electrical properties tomography: an in vivo study. *Brain topography*, 34, 1, 56–63.

[5] Hasgall P.A., Di Gennaro F., C. Baumgartner, E. Neufeld, B. Lloyd, M.C. Gosselin, D. Payne, A. Klingenböck, and N. Kuster. 2018. IT'IS database for thermal and electromagnetic parameters of biological tissues. Version 4.0. (2018). DOI: 10.13099/VIP21000-04-0.

[6] W.P. Segars, B.M.W. Tsui, J. Cai, F.-F. Yin, G.S.K. Fung, and E. Samei. 2018. Application of the 4-D XCAT phantoms in biomedical imaging and beyond. *IEEE Transactions on Medical Imaging*, 37, 3, 680–692.

[7] Andrzej J. Surowiec, Stanislaw S. Stuchly, J. Robin Barr, and Arvind Swarup. 1988. Dielectric properties of breast carcinoma and the surrounding tissues. *IEEE Transactions on Biomedical Engineering*, 35, 4, 257–263.

[8] B.A. Wilkinson, Rod Smallwood, A. Keshtar, J. A. Lee, and F.C. Hamdy. 2002. Electrical impedance spectroscopy and the diagnosis of bladder pathology: a pilot study. *The Journal of urology*, 168, 4, 1563–1567.

[9] Xiaotong Zhang, Jiaen Liu, and Bin He. 2014. Magnetic-resonance-based electrical properties tomography: a review. *IEEE Reviews in Biomedical Engineering*, 7, 87–96. DOI: 10.1109/RBME.2013.2297206.

Library for Feature Calculation in the Context-Recognition Domain

Vito Janko
Jožef Stefan Institute
Department of Intelligent Systems
Ljubljana, Slovenia
vito.janko@ijs.si

Matjaž Boštic
Jožef Stefan Institute
Department of Intelligent Systems
Ljubljana, Slovenia
bosticmatjaz@gmail.com

Junoš Lukan
Jožef Stefan Institute
Department of Intelligent Systems
Ljubljana, Slovenia
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
junos.lukan@ijs.si

Gašper Slapničar
Jožef Stefan Institute
Department of Intelligent Systems
Ljubljana, Slovenia
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
gasper.slapnicar@ijs.si

ABSTRACT

Context recognition is a mature artificial intelligence domain with established methods for a variety of tasks. A typical machine learning pipeline in this domain includes data preprocessing, feature extraction and model training. The second of these steps is typically the most challenging, as sufficient expert knowledge is required to design good features for a particular problem. We present a Python library which offers a simple interface for feature calculation useful in a myriad of different tasks, from activity recognition to physiological signal analysis. It also offers additional useful tools for data preprocessing and machine learning, such as a custom wrapper feature selection method and prediction smoothing using Hidden Markov Models. The usefulness and usage is demonstrated on the 2018 SHL locomotion challenge where a few simple lines of code allow us to achieve solid predictive performance with F1 score of up to 93.1, notably surpassing the baseline performance and nearing the results of the winning submission.

KEYWORDS

feature calculation, python library, context recognition, machine learning

1 INTRODUCTION

Context recognition is a vague term encompassing a variety of tasks where sensors are put on (or around) a person and are then used to determine something about them. For example, sensors in a smartphone can determine if a user is standing, walking, running or even falling. A wristband sensor can read physiological signals like heart-rate or sweating to determine stress or blood pressure. These kinds of applications are usually used for self monitoring in sport activities or for helping the users manage various medical conditions.

The context-recognition field is quite mature and its applications often come pre-installed in many commercial devices like wristbands and smartphones. Nonetheless, the development of a new context recognition system can be tedious and time-consuming. It usually consists of collecting relevant sensor data, parsing it to a suitable format, calculating features based on this data and finally training the model.

In this work we present a Python library focused on streamlining this process. Its main functionality is calculating the features from sensor data. It can generate over a hundred different features that have proven themselves in various context-recognition projects we tackled in the past [4, 3, 5]. Loosely, the features can be divided in two categories: those suitable for motion data (e.g. generated by accelerometer or gyroscope) and those specialized for physiological signals.

Furthermore, the library implements some other functionalities that are often used in context recognition pipelines: reshaping data into windows, re-sampling the data, selecting the best features after generating them and a method for smoothing the final predictions of the classifier using a Hidden Markov Model approach.

To demonstrate the usefulness of the library we used its functionalities exclusively (with the exception of a generic Random Forest classifier [11]) on the SHL Challenge dataset [16]. We demonstrate the whole pipeline, from reading in the raw data to the finished context-recognition system that is comparable to the best-performing submissions in the SHL Challenge.

2 LIBRARY FUNCTIONALITIES

The library is implemented in Python as this has been the most popular data science language in recent years [6]. It is available in a public repository with `pip install cr-features` command.

Its main and most valuable functionality lies in feature generation. The 'motion features' are listed in Section 2.1, while the 'physiological features' are described in Section 2.2. Remaining non-feature related functionalities are explained in Section 2.3.

2.1 Motion Sensors Features

Features listed in the first two subsections are general and can be applied on any sensor data time-series. The last subsection (2.1.3), on the other hand, lists features that have an additional semantic interpretation for acceleration and require data from three (x,y,z) axes. The library defines similar sensor subsets for some other sensors (e.g. gyroscope). Only a subset of features is listed for brevity, while the full list can be found in the documentation [1].

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

2.1.1 General Statistical Features.

- Basic statistical measures: maximum, minimum, standard deviation, median, mean difference between samples.
- Number of peaks – useful for detecting and counting steps, estimating the energy expenditure and determining the frequency of motion: peak count, number of times data crossed its mean value, longest time data was above or below its mean value.
- Different data aggregations that can indicate the intensity of the activity: (squared) sum of values, sum of absolute values.
- Autocorrelations (i.e. how similar the data is to a shifted version of itself) which indicate periodicity: autocorrelation for raw data, for peak positions, for mean crossings.
- Data shape: skewness (a measure of symmetry, or more precisely, the lack of symmetry), kurtosis (a measure of whether the data is heavy-tailed or light-tailed relative to a normal distribution), interquartile range.

2.1.2 Frequency Features. They are calculated by first computing an estimate of the power spectral density of the signal via a periodogram. We used the Welch's method which is an improvement over the traditional methods in that it reduces noise in the estimated power spectra.

Once the periodogram is obtained, the following features are computed: the magnitude value of the three highest peaks in periodogram, the three highest frequencies corresponding to the highest peaks, energy of the signal calculated as the sum of squared FFT component magnitudes, entropy of the signal computed as the information entropy of the normalized FFT component magnitudes, and the distribution of the FFT magnitudes into 10 equal sized bins ranging from 0 Hz to $F_s/2$, where F_s is the sampling frequency. Finally, we also computed the previously described skewness and kurtosis for the periodogram.

Most of the described features are useful for finding different periodic patterns, how often they occur and how intense they are.

2.1.3 Accelerometer Features.

- Phone rotation estimation. First, roll and pitch are calculated, then we calculate their characteristics: mean, standard deviation, peaks, autocorrelations.
- Physical interpretations: velocity, kinetic energy.
- Comparing data axis; useful for determining the sensor orientation relative to the direction of motion: correlation between axis data, comparing their means, mean direction of the vector they form.

2.2 Physiological Features

Physiological features are useful for obtaining information about a person's physiological state, typically reflected in their cardiovascular response. We computed several features from signals obtainable from many modern wristbands as described in the sections below.

2.2.1 Heart Rate and Heart Rate Variability. Cardiovascular measures are widely used to predict both medical problems as well as psychological processes [7]. They range from simple heart rate calculations to more complex heart rate variability indicators. Heart rate variability is a measure of how quickly heart rate itself changes and it is usually calculated on a beat-by-beat basis, considering the inter-beat interval (IBI). It reflects the interaction between sympathetic and parasympathetic regulation of heart beat [10] and is thus an especially useful physiological indicator.

Calculation of features related to cardiovascular activity follows recommendations by Malik, Bigger, Camm, Kleiger, Malliani, Moss and Schwartz [8]. To describe heart rate variability, the Fourier transform of inter-beat intervals is calculated and then several frequency features are derived from the spectrum [5].

2.2.2 Skin Conductivity. Electrical conductivity of the skin varies due to physiological changes in sweat glands, which are controlled by the autonomic nervous system. In a simple model of resistive properties of skin and sweat glands, whenever the level of sweat in the glands is increased, its conductivity also increases [2]. Sweat glands thus act as variable resistors and actual sweating, that is sweat secretion from the glands, is not needed for this change to be measurable.

Changes in skin conductivity are not only triggered by other physiological changes, such as the ones in (skin) temperature, but also reflect psychological processes. Skin conductivity can indicate cognitive activity or emotional responses and can do so with good sensitivity [see 7, for an exhaustive review].

Sweat glands continuously adapt to their environment and their reactions can be slow or fast. Two main modes of fluctuations are therefore distinguished: skin conductance level changes, which are slow variations of the general trend, also called tonic electrodermal measures, and skin conductance responses, quick reactions, also called phasic electrodermal measures [13].

To calculate skin conductivity features the two components are first separated. This is done using the EDA Explorer library [14] which enables searching for peaks (SCRs) in the signal by specifying their desired characteristics.

The signal is first filtered using a Butterworth low-pass filter from SciPy [15]. Next, the peaks are detected by considering their amplitude, onset, and offset time.

Once the SCRs are found, their characteristics are calculated which can be used as features. These include their number and rate (relative frequency in time) as well as the means and maxima of various characteristics, such as their maximum amplitude, their duration, increase time etc.

The tonic component is calculated using `peakutils` [9]. It is detected as the signal baseline, fitting a 10-th degree polynomial to the signal. Similarly to the phasic component, statistical features are calculated, such as the difference between this component and the raw signal, and the sum of its derivative.

2.2.3 Skin Temperature. Skin temperature is a fairly simple physiological parameter, both from the point of view of measurement as well as feature calculation. It can still serve as an indicator of affect [7]. Unlike the other physiological parameters which make use of expert features only some generic statistical features are calculated for this indicator.

2.3 Other Functionalities

The following functionalities are not directly related to the feature generation but are nonetheless often used in conjunction with it – and can thus make the workflow more straightforward.

2.3.1 Resize, Resample. The presented library works with raw data in matrix form: each row representing one window of data, i.e. one instance. If the original data is in the form of 1D timeseries, the `convertInputInto2d` function can reformat it in the required format. It can work both with windows of fixed number of data samples as well as windows representing a fixed time interval. Another frequent pre-processing step is down-sampling the data and it can be done with the `resample` function.

2.3.2 Wrapper Feature Selection. While many feature selection libraries already exist (e.g. scikit-learn [11]), we implemented another one in this library as it was frequently used in our previous work [4, 3]. It combines the relatively common ‘wrapper’ approach with reducing the feature count using correlations. It works in three steps:

(1) Calculate the information gain for every feature and rank them based on it.
(2) Calculate the correlation between each feature pair. If the correlation exceeds the given threshold, discard the one with lower information gain.
(3) Create the classifier using only the highest ranking feature and measure the accuracy using a validation set. Then add the second feature and measure the accuracy again. If it was the same or higher, keep the feature, otherwise discard it. Repeat for all other remaining features.

2.3.3 Hidden Markov Model Smoothing. The final functionality is a tool to post-process the predictions of the context-recognition system – taking into account the temporal dependencies between the instances.

Take an example in which the classifier predicts the following minute-by-minute sequence: ‘subway’, ‘subway’, ‘bus’, ‘subway’, ‘subway’. It is far more likely that the ‘bus’ prediction is a misclassification than switching vehicles for just a minute.

Such a sequence can be corrected using a Hidden Markov Model (HMM). This model assumes that there are hidden states corresponding to real activities which emit visible signals – classifications. The parameters of this models can be inferred from the matrix of transitions probabilities and confusion matrix of the predictor.

Once the parameters are estimated, the Viterbi algorithm is used in the background to determine the most likely sequence of hidden states (activities) given visible emissions (predictions). In many domains [4, 12] this method significantly improves the final prediction accuracy.

While this method is least connected to the feature generation, we have not seen it implemented in a different library and have found it greatly useful.

3 USAGE EXAMPLE

We illustrate the usage of our library with an example: The Sussex-Huawei Locomotion Challenge 2018 [16]. This was a worldwide open activity recognition challenge with monetary incentives, organized as part of the HASCA workshop within UbiComp conference. 17 teams participated with 19 submissions. The goal was to train a recognition pipeline on the provided training data and then use it to classify the withheld test data as well as possible in terms of the F_1 score metric.

3.1 SHL Dataset

The challenge used a subset of the full dataset which was recorded over a period of 7 months by 3 participants engaging in 8 different modes of transportation (still, walk, run, bike, car, bus, train and subway). The phones were worn on 4 body positions, namely the hand, torso, hip pocket and in a backpack and recorded 16 sensor modalities simultaneously. This totalled to 2812 hours of labelled data and this is considered one of the largest such datasets openly available [16].

In the actual challenge, the subset used was the data recorded by one of the three participants, which included 82 days of recording, split into the training set (271 hours) and testing set (95 hours). Raw data from 7 sensors was provided: accelerometer, gyroscope, magnetometer, linear acceleration, gravity, orientation and air pressure. All were sampled at 100 Hz [16].

Data was split into 1-minute segments using a sliding window without overlap and then randomly shuffled, providing consistent instances. Finally, the training data had 16 310 such instances and test data had 5698, where each instance contained 6000 samples. This highlights the sheer size of the data and the challenges in processing it in full.

3.2 Methods

We used a traditional ML pipeline for this task: first preprocessing the data, then computing informative features, selecting the best of them and finally using them to train and evaluate a classification model. We added another not so traditional step: smoothing the predictions using HMM.

All steps except training and evaluation were done in few lines using the presented library; the Python code (with some missing steps in comments) is given below. All classification was done using scikit-learn implementation of Random Forest with default parameters.

```
from CalculatingFeatures import resample,
    calculateFeatures, selectFeatures, hmm_smoothing

# Data was already windowed
# Data was resampled from 100 Hz to 20 Hz
acc_x = pd.read_csv(path, sep=" ")
acc_x = resample(acc_x, 6000, 1200)
# Repeat for all data types (and axes)
features_train = calculateFeatures(
    acc_x,
    acc_y,
    acc_z,
    featureNames=accelerationNames,
    prefix="acc",
)
# Repeat for all data types and train/test/valid sets
# Merge in one dataframe
selected = selectFeatures(
    features_train, features_validation
)
f1, cf, predictions = evaluate(
    features_train[selected],
    features_test[selected],
    labels_train,
    labels_test,
)
smoothed = hmm_smoothing(labels_train, cf, predictions)
# smoothed is an array representing final output
```

3.3 Results

We compared the results – in terms of F_1 score – of different stages in the machine learning pipeline against the top three submissions in the competition.

In the first stage we used just the mean and standard deviation as features (and calculated them for each data modality) to provide a baseline solution. Next, we calculated some features using the presented library. We then selected only a subset of them

Table 1: A comparison of different versions of the pipeline, against the best submissions in the SHL Challenge. The number of features used in our methods is also listed.

Experiment	# features	F_1 score
Baseline	38	80.3
All features	298	87.7
Feature selection	130	87.1
HMM	130	93.1
Third place	/	87.5
Second place	/	92.4
First place	/	93.9

and again measured the performance. Finally, we used the HMM smoothing; a post-processing step described in Section 2.3.3.

Results are shown in Table 1. It shows that the features generated by the library substantially improve the performance. The feature selection, on the other hand – while significantly reducing the number of features required – did not increase performance. Of note, the performance did increase in the internal validation set, but this gain did not translate to the test set. The final jump in performance was achieved using the HMM smoothing and we highly recommend this method in this and similar domains.

Using just the methods in the presented library and no parameter or method tuning we achieved the results comparable with the first placed submission to the challenge.

4 CONCLUSION

In this paper we demonstrated the base usage of a Python library capable of calculating features suitable for the context recognition domain. The most important features that can be calculated are listed in this paper with specialized ones thoroughly described.

We also showed on a topical example (SHL Challenge dataset) how only a few lines of code can generate a very capable context-recognition system that can compete with the best entries submitted to this challenge. Such system can be improved with extensive tuning but we provide a solid starting point.

It is our hope that by making this library publicly available we can help the workflow of many future context-recognition researchers.

ACKNOWLEDGMENTS

We acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0209).

REFERENCES

[1] Matjaž Boštic, Vito Janko, Gašper Slapničar, Jakob Valič and Junoš Lukan. 2021. cr-features. A library for feature calculation in the context-recognition domain. https://repo.ijs.si/matjazbostic/calculatingfeatures. Accessed: 2021-09-20. (2021).

[2] Wolfram Boucsein. 2012. *Electrodermal activity*. Springer Science & Business Media.

[3] Božidara Cvetković, Robert Szeklicki, Vito Janko, Przemyslaw Lutomski and Mitja Luštrek. 2018. Real-time activity monitoring with a wristband and a smartphone. *Information Fusion*, 43, 77–93.

[4] Martin Gjoreski, Vito Janko, Gašper Slapničar, Miha Mlakar, Nina Reščič, Jani Bizjak, Vid Drobnič, Matej Marinko, Nejc Mlakar, Mitja Luštrek et al. 2020. Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. *Information Fusion*, 62, 47–62.

[5] Martin Gjoreski, Mitja Luštrek, Matjaž Gams and Hristijan Gjoreski. 2017. Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73, 159–170. DOI: 10.1016/j.jbi.2017.08.006.

[6] Harshil. 2021. Tools of the trade: a short history. https://www.kaggle.com/haakakak/tools-of-the-trade-a-short-history/. Accessed: 2021-09-20. (2021).

[7] Sylvia D. Kreibig. 2010. Autonomic nervous system activity in emotion: a review. *Biological Psychology*, 84, 3, 394–421. DOI: 10.1016/j.biopsycho.2010.03.010.

[8] M. Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A. Malliani, A. J. Moss and P. J. Schwartz. 1996. Heart rate variability: standards of measurement, physiological interpretation, and clinical use. *European Heart Journal*, 17, 3, 354–381. DOI: 10.1093/oxfordjournals.eurheartj.a014868.

[9] Lucas Hermann Negri and Christophe Vestri. 2017. Lucashn/peakutils: v1.1.0. (2017). DOI: 10.5281/ZENODO.887917.

[10] M Pagani, F Lombardi, S Guzzetti, O Rimoldi, R Furlan, P Pizzinelli, G Sandrone, G Malfatto, S Dell'Orto and E Piccaluga. 1986. Power spectral analysis of heart rate and arterial pressure variabilities as a marker of sympatho-vagal interaction in man and conscious dog. *Circulation Research*, 59, 2, 178–193. DOI: 10.1161/01.res.59.2.178.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[12] Clément Picard, Vito Janko, Nina Reščič, Martin Gjoreski and Mitja Luštrek. 2021. Identification of cooking preparation using motion capture data: a submission to the cooking activity recognition challenge. In *Human Activity Recognition Challenge*. Springer, 103–113.

[13] Society for Psychophysiological Research Ad Hoc Committee on Electrodermal Measures. 2012. Publication recommendations for electrodermal measurements. *Psychophysiology*, 49, 8, 1017–1034. DOI: 10.1111/j.1469-8986.2012.01384.x.

[14] Sara Taylor, Natasha Jaques, Weixuan Chen, Szymon Fedor, Akane Sano and Rosalind Picard. 2015. Automatic identification of artifacts in electrodermal activity data. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. DOI: 10.1109/embc.2015.7318762.

[15] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, ..., Paul van Mulbregt and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. DOI: 10.1038/s41592-019-0686-2.

[16] Lin Wang, Hristijan Gjoreski, Kazuya Murao, Tsuyoshi Okita and Daniel Roggen. 2018. Summary of the sussex-huawei locomotion-transportation recognition challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1521–1530. DOI: 10.1145/3267305.3267519.

Določanje slikovnega prostora na umetniških slikah

Reconstruction of image space depicted on artistic paintings

Nadezhda Komarova
Gregor Anželj
nadezhdakomarova7@gmail.com
gregor.anzelj@gmail.com
Gimnazija Bežigrad
1000 Ljubljana, Slovenia

Borut Batagelj
Narvika Bovcon
Franc Solina
borut.batagelj@fri.uni-lj.si
narvika.bovcon@fri.uni-lj.si
franc.solina@fri.uni-lj.si
Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

POVZETEK

V članku poročamo o analizi slikovnega prostora na umetniških slikah s pomočjo metod računalniškega vida. Naš cilj je bil, da ugotovimo, ali je možno zgolj na osnovi zaznave obrazov na slikah določiti prostorsko organizacijo slike. Analiza je potekala na izbranem vzorcu 3356 slik. Najprej smo določili tridimenzionalne koordinate zaznanih obrazov na posamezni sliki. Nato smo tem točkam priredili ravnino. Slikovni prostor smo tako določili z enačbo prirejene ravnine oziroma kotom med to ravnino in slikovno ravnino. Bolj kot je ravnina, ki jo določajo obrazi, nagnjena od navpične smeri, globlji je prikazani slikovni prostor.

KLJUČNE BESEDE

računalniški vid, slikovni prostor, zaznava obrazov, umetnostna zgodovina

ABSTRACT

In the article, we report on the analysis of the image space depicted on artistic paintings utilizing methods of computer vision. Our aim was to find out whether one can recover the spatial organization of a picture based on detection of faces. The analysis was conducted on the sample of 3356 paintings. First, 3D coordinates of faces were determined. Then, a plane was fitted to the faces on every painting. Images were therefore described in terms of the angle between the fitted plane and the picture plane. The bigger the angle between both planes, the deeper the picture space depicted.

KEYWORDS

computer vision, image space, face detection, art history

1 UVOD IN MOTIVACIJA

Odločili smo se povezati dve raziskovalni področji, ki sta si navidez zelo vsaksebi, to je umetnostna zgodovina in umetna inteligenca. Metode računalniškega vida se že redno uporabljajo tudi za analizo umetniških slik [12]. Večina teh raziskav je osredotočena na analizo posameznih ali manjšega števila umetniških slik. Po drugi strani smo danes v dobi velepodatkov (angl. *Big Data*), saj je vedno več informacij dostopnih v digitalni obliki. Tudi velike zbirke reprodukcij umetniških slik so danes prosto dostopne na medmrežju, na primer Google Arts and Culture, Wikimedia Commons, Getty Open Content Program, ADA (Archive of Digital Art) in druge [4]. Z analizo in vizualizacijo velikih umetniških zbirk se je prvi začel ukvarjati Lev Manovich [8]. Leta 2012 je preučeval vizualizacijske metode za družboslovne vede in medijske raziskave. Ukvarjal se je z informativno, uporabno in estetsko vrednostjo vizualizacij [9].

Analiza razlik med predstavitvijo prostora s fotografijo in umetniško sliko je bila narejena leta 2014 [11]. S statistično analizo slik tihožitij, ki so jih ustvarili udeleženci eksperimenta, so ugotovili, da so predmeti, na katere so udeleženci usmerjali pozornost, naslikani večji kot so na fotografijah. Zato je vprašanje, ali je dosledna uporaba linearne perspektive najbolj primerna metoda za posnemanje sveta [1]. Umetnostna zgodovina nam nazorno prikaže, da so umetniki za posnemanje sveta uporabljali zelo različne pristope.

Pri naši analizi slikovnega prostora smo izhajali iz dveh predpostavk:

(1) v raziskavi želimo analizirati veliko število umetniških slik v smislu današnjega trenda *Big Data*,
(2) uporabiti želimo take metode računalniškega vida, ki delujejo hitro in čimbolj zanesljivo.

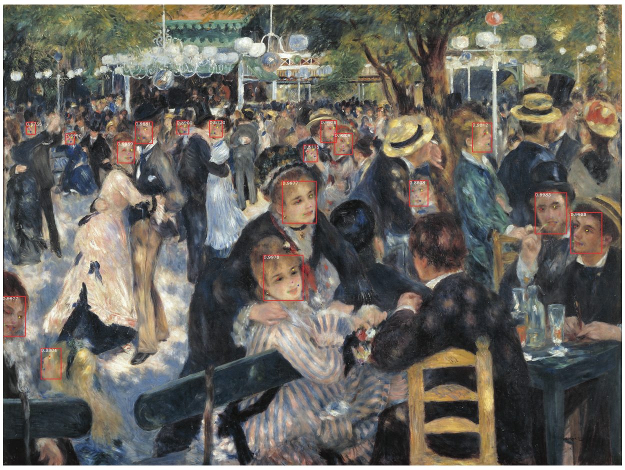
Med hitre in zanesljive metode računalniškega vida zagotovo sodi zaznava in identifikacija oseb na osnovi njihovih obrazov. Zaradi varnostnih razlogov se je teh problemov na področju biometrije lotilo že zelo veliko znanstvenikov. Danes obstajajo hitre in zanesljive metode za zaznavo in analizo obrazov na slikah [10].

Za navdih nam je služil članek Irvinga Zupnicka iz leta 1959 [14], objavljen še veliko pred uporabo računalnikov v likovni umetnosti, ki opisuje kako je na slikah iz različnih umetnostnih obdobjih organiziran slikovni prostor. Zato smo si zastavili vprašanje, ali je mogoče s pomočjo metod računalniškega vida rekonstruirati slikovni prostor na umetniških slikah? Bolj konkretno, ali ga je mogoče rekonstruirati na osnovi zaznave obrazov na slikah? Določitve 3D razsežnosti prostora, upodobljenega na sliki, smo se lotili na osnovi pozicije obrazov na sliki (x in y koordinate) in njihove velikosti, kar nam daje grobo informacijo o tretji dimenziji z — to je oddaljenosti obraza od opazovalca. Ta pristop seveda temelji na predpostavki, da so na slikah ljudje oziroma da so upodobljeni njihovi dovolj veliki obrazi. Resda v zgodovini likovne umetnosti poznamo veliko tihožitij ali pokrajinskih slik, na katerih ni obrazov. Toda velika večina umetniških slik iz obdobja pred izumom fotografije dejansko upodablja ljudi oz. njihove obraze.

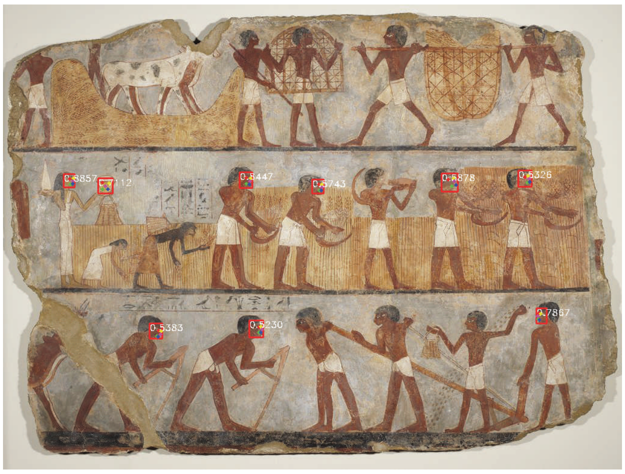
Iz javno dostopnih baz umetniških slik smo za našo študijo izbrali testno množico 3356 slik iz različnih umetnostnozgodovinskih obdobij in žanrov.

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

2 SLIKOVNI PROSTOR NA UMETNIŠKIH SLIKAH



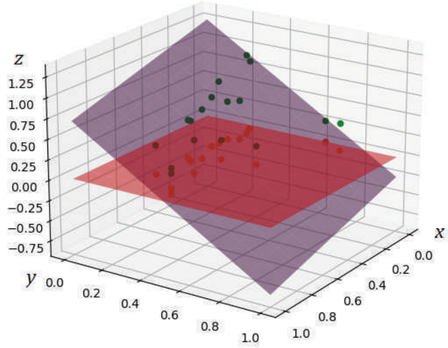
Slika 1: **Auguste Renoir, *Ples v Le Moulin de la Galette*; vidijo se zaznani obrazi. Velikost obrazov jasno odraža globino slikarskega prostora.**



Slika 2: Poslikava v grobnici *Unsu*. Vsi obrazi se enake velikosti, ves slikarski prostor je zgoščen kar v ravnini poslikave.

Vsakemu obrazu na slikah smo priredili tridimenzionalne koordinate, ki pa niso bile zanesljive v absolutnih vrednostih, temveč odražajo zgolj relativne razdalje. Nato smo tem obraznim točkam priredili ravnine s smislu vsote najmanjših kvadratov razdalj med točkami in iskano ravnino. Pri tistih slikah, ki prikazujejo obraze, ki so v spodnjem delu slike opazovalcu blizu, in se višje na sliki postopno oddaljujejo (Slika 1), so dobljene ravnine bolj nagnjene v globino kot pri tistih, kjer so vsi obrazi približno na enaki razdalji od opazovalca (Slika 2). V takih primerih je dobljena ravnina skorajda vzporedna s površino slike. Rafaelova *Atenska šola* in staroegipčanska poslikava v grobnici *Unsu* imata zelo različni prostorski ureditvi. Na prvi sliki se obrazi zmanjšujejo z oddaljevanjem ljudi. Ravnina, prirejena točkam na tej sliki, je zato nagnjena v globino (Slika 3).

Po drugi strani tudi poslikava na Sliki 2 prikazuje množico ljudi, vendar so vsi enake višine in njihovi obrazi so enako veliki. Ravnina, prirejena obrazom na egipčanski sliki, je zato vzporedna ravnini $z = 0$. Za egipčansko slikarstvo je značilno konceptualno upodabljanje prostora: velikosti oseb niso določene s prostorskim oddaljevanjem, temveč npr. z družbenim statusom.



Slika 3: Vijoličasta ravnina, ki se prilega 3D pozicijam obrazov na Renoirjevem *Plesu v Le Moulin de la Galette* in in rdeča ravnina $z = 0$ – ploskev slikarskega platna, na kateri smo zaznali obraze.

3 ZAZNAVA OBRAZOV

Predpostavili smo, da so resnični obrazi pri vseh osebah približno enako veliki. Zato so bili večji obrazi obravnavani kot bližji površini slike in manjši kot bolj oddaljeni od površine slike oz. od opazovalca.

Zaznani so bili z orodjem *RetinaFace*, ki izvede dvodimenzionalno poravnavo in tridimenzionalno rekonstrukcijo obraza [2]. Zasnovan je na osnovi globoke nevronske mreže.

Detektor vrne podatke o obrazih v dvodimenzionalnem prostoru površine slike, torej imajo središča obraznih okvirjev in točke oči, nosu ter ust samo x in y koordinate. Toda za rekonstrukcijo tridimenzionalnega prostora slike potrebujemo tudi globine obrazov oz. koordinato z. Tridimenzionalni prostor, kot ga prikazuje umetniška slika, se razlikuje od fotografskega predvsem zato, ker slikarji redko dosledno upoštevajo linearno perspektivo. Na fotografijah je perspektiva po drugi strani bolj konsistentno določena. Zato je na njih mogoče z enačbo (1) [6] določiti oddaljenost predmeta od kamere:

$$d = \frac{f \cdot h_r \cdot h}{h_i \cdot h_s} \tag{1}$$

Z enačbo (1) izračunamo oddaljenost d objekta v milimetrih, če je f goriščna razdalja fotoaparata, h_r resnična višina objekta v milimetrih, h višina slike v pikslih, h_i višina objekta na sliki v pikslih in h_s višina senzorja fotoaparata v milimetrih. Z njo so bile določene tudi oddaljenosti obrazov na slikah v vzorcu, pri čemer so bile uporabljene vrednosti goriščne razdalje in višine senzorja, kvocient katerih opiše, kako vidijo človeške oči. Četudi je bilo po tem postopku nemogoče določiti natančne tridimenzionalne koordinate obrazov na sliki, so bile določene relativne oddaljenosti med obrazi in površino slike. Za namen te raziskave tudi niti ni pomembno, če zaznamo vse obraze na sliki.

4 GEOMETRIJSKA INTERPRETACIJA PROSTORA

Parametre A, B in C enačbe ravnine $z = Ax + By + C$ smo določili z minimizacijo funkcije

$$E(A, B, C) = \sum_{i=1}^{m} (Ax_i + By_i + C - z_i)^2, \tag{2}$$

kjer m pomeni število točk in x_i, y_i ter z_i koordinate točk. Funkcija (2) doseže minimum, ko je $\nabla E = (0, 0, 0)$ [3]. Za gradient te funkcije velja $\nabla E = (\frac{\partial E}{\partial A}, \frac{\partial E}{\partial B}, \frac{\partial E}{\partial C})$, kjer so $\frac{\partial E}{\partial A}$, $\frac{\partial E}{\partial B}$ in $\frac{\partial E}{\partial C}$ naslednji.

$$\frac{\partial E}{\partial A} = 2\sum_{i=1}^{m} x_i(Ax_i + By_i + C - z_i) \tag{3}$$

$$\frac{\partial E}{\partial B} = 2\sum_{i=1}^{m} y_i(Ax_i + By_i + C - z_i) \tag{4}$$

$$\frac{\partial E}{\partial C} = 2\sum_{i=1}^{m} (Ax_i + By_i + C - z_i) \tag{5}$$

Tako množici 3D točk priredimo ravnino z minimizacijo razdalj med temi točkami in njihovimi slikami na ploskvi v smeri z. Koeficienti A, B in C so zato rešitve sistema linearnih enačb (6), (7) in (8).

$$A\sum_{i=1}^{m} x_i^2 + B\sum_{i=1}^{m} x_i y_i + C\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x_i z_i \tag{6}$$

$$A\sum_{i=1}^{m} x_i y_i + B\sum_{i=1}^{m} y_i^2 + C\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} y_i z_i \tag{7}$$

$$A\sum_{i=1}^{m} x_i + B\sum_{i=1}^{m} y_i + C = \sum_{i=1}^{m} z_i \tag{8}$$

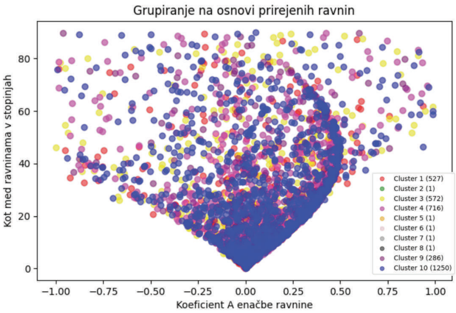
5 REZULTATI

Slike smo izbrali iz prostodostopne zbirke WikiArt (https://www.wikiart.org), kjer so umetnine med drugim razdeljene po žanrih. Izbrana so bila slikarska dela (potrebno je bilo izločiti npr. kiparska), kjer je bilo upodobljenih več ljudi. Iz zbirke WikiArt so bila zato izbrana dela iz žanrov *pastorale* (77 slik), *allegorical painting* (1225 slik), *history painting* (1377 slik) in *literary painting* (667 slik), in sicer skupaj 3356 slik. Poleg žanra smo imeli tudi podatke o umetnostno zgodovinskem obdobju v katero sodi posamezna slika. Zanimalo nas je, kako lahko le na osnovi teh podatkov smiselno razdelimo testno množico slik z metodo gručenja in ali je ta delitev relevantna z vidika umetnostno zgodovine.

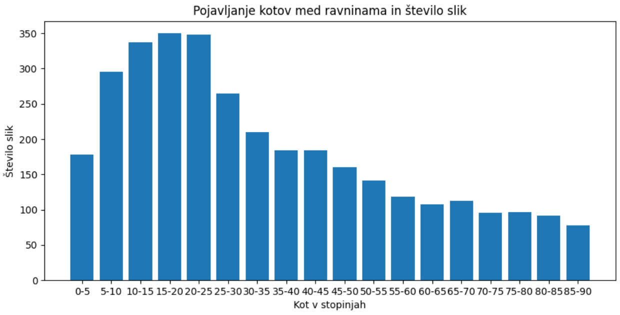
Kot kriterij pri gručenju so bile uporabljene enačbe ravnin ter kot med prirejeno ravnino in slikovno ravnino $z = 0$. Detektor *RetinaFace* opiše slednje s tremi parametri – rotacijami okoli osi x, y in z (v pozitivni in negativni smeri). Pri posamezni sliki so bile izbrane rotacije v vsaki smeri z največjimi absolutnimi vrednostmi.

Gručenje je bilo opravljeno z algoritmom BIRCH, implementiranim s knjižnico *scikit-learn*. BIRCH (angl. *Balanced Iterative Reducing and Clustering using Hierarchies*) je algoritem gručenja, ki je posebej prilagojen delu z večjimi podatkovnimi vzorci [7].

Na Sliki 4 so ekstremne vrednosti izločene. Prikazana je razporeditev slik po gručenju na osnovi ravnin. Bila je izvedena primerjava tega, katerim umetnostnim slogom pripadajo slike v posameznih razredih. To je bilo mogoče, saj je bila vsaka slika v zbirki označena poleg žanra tudi z letom nastanka in umetnostnim slogom (barok, romantika ipd.). Število razredov smo omejili na deset. Zaradi izrazite drugačnosti prostorske razporeditve na nekaterih slikah so bile slednje izločene v posamezne razrede (2, 5, 6, 7 in 8). Ti razredi vsebujejo le po eno sliko in niso vidni na Sliki 4.



Slika 4: Razporeditev razredov pri gručenju na osnovi ravnin. Gruče so razpršene in izrazite razmejitve med njimi ni.



Slika 5: Zastopanost posameznih kotov za slike v testni množici.

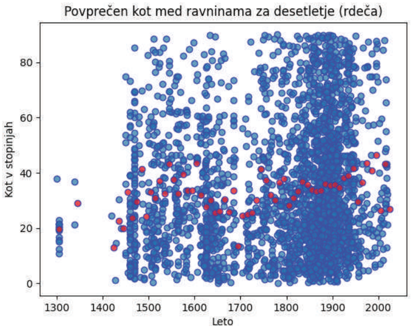
Histogram na Sliki 5 prikazuje zastopanost različnih intervalov kotov v proučevanem vzorcu. Vidi se, da je bil največji delež slik takih, kjer je bil kot med ravninama med 15 in 20 stopinj, kar se zdi relativno malo. Večji koti med ravninama večinoma ustrezajo slikam, kjer se upodobljene osebe enotno oddaljujejo oz. približujejo. Če je bil kot med ravnino, ki je bila prirejena obrazom na sliki, in ravnino $z = 0$ izračunan kot natančno 0 stopinj, je to pomenilo, da na sliki ni bilo zaznanega nobenega obraza, samo en obraz ali pa so imeli vsi obrazi enake globine. Na intervalu od 0 do 5 stopinj (Slika 5) je bil najpogostejši barok, na preostalih intervalih po romantika. Ni pa na nobenem intervalu močno prevladoval le en slog, saj je odstotek slik, ki je pripadal najpogostejšemu slogu v posameznem intervalu med 20 in 30%.

Za določitev korelacije med časom nastanka posamezne slike in kotom med ravninama za to sliko je bil uporabljen Spearmanov koeficient korelacije. Ta predstavlja neparametersko stopnjo povezanosti med spremenljivkama oz. kako dobro je mogoče opisati njun odnos z monotono funkcijo [13]. Koeficient je bil 0.183, kar predstavlja šibko pozitivno korelacijo. p vrednost je bila v tem primeru blizu 0, kar pomeni, da korelacija med letom nastanka slike in kotom, ki odraža slikarsko globino ni linearna. Na prikazu na Sliki 6 je razvidno, da če opazujemo obdobje od približno leta 1700 in vse do danes, povprečen kot med ravninama za posamezna desetletja blago narašča.

6 RAZPRAVA

Glavna hipoteza naše raziskave je bila, ali lahko na nek enostaven način ugotovimo kakšen je slikarski prostor, to je, kako izrazita je globinska dimenzija na dani umetniški sliki. Slikarski prostor pa je povezan tako z umetnostno zgodovinskim obdobjem v katerega sodi slika, kot tudi z žanrom slike. Na ta način se nam odpira možnost avtomatske klasifikacije velikega števila slik, bodisi s statističnimi metodami, še bolj pa bi prišle v poštev metode strojnega učenja.

Odločili smo se, da bomo slikovni prostor določali posredno s pomočjo zaznave obrazov. Ko je bil posamezen obraz zaznan z orodjem *RetinaFace*, je bil s tem določen obrazni okvir na določeni



Slika 6: Koti med ravninama v odvisnosti od časa nastanka slike. Rdeče točke predstavljajo povprečen kot za posamezno desetletje.

koordinati x in y na ravnini slike. Velikost obraznega okvirja pa nam je dal še informacijo o relativni oddaljenosti obraza z od ravnine slike. Zanesljivost zaznave obrazov na umetniških slikah je bil verjetno nekoliko slabši, saj je bil *RetinaFace* naučen na fotografijah obrazov in ne na umetniških upodobitvah [2]. V kakšni prihodnji raziskavi bi lahko uporabili še dodatne informacije, ki jih daje orodje *RetinaFace* za zaznavo obrazov: orientacija obraza, lega oči, nosu in ust, spol ter starost osebe, določeno na osnovi obraza. Poleg tega bi lahko v prihodnjih raziskavah pri analizi slik upoštevali tudi barvno sestavo in druge slikovne značilke, ki jih lahko robustno določimo z metodami računalniškega vida [12]. Sami smo se ukvarjali npr. z detekcijo črt perspektivne projekcije na fotografijah [5, 1].

Četudi smo v našem preizkusu metode likovna dela združevali v razrede po podobnosti prostorske ureditve, se niso pokazale stroge meje med umetnostnimi slogi slik. Informativna pa je bila korelacija med časom nastanka dela in kotom med ravninama. V izbranem vzorcu slik različni umetnostnozgodovinski slogi niso bili povsem enakomerno zastopani in je bilo npr. veliko del iz romantike. Za vsako zgodovinsko obdobje so najverjetneje izrazite določene medsebojne povezanosti teh značilnosti. Ustaljen umetnostnozgodovinski pristop pri analizi slik je sočasno opazovanje dveh ali več del, pri katerih raziskovalec na osnovi svojega predhodnega znanja izloči značilne poteze, razlike ipd. [8]. Strojno učenje bi na tej točki postalo učinkovito, saj po eni strani nudi možnost analize velike količine podatkov, odkrivanje sočasnih povezav med različnimi značilkami, po drugi strani pa zagotavlja objektivnost matematičnih pristopov. Zato bi bilo v nadaljevanju koristno uporabiti poleg obrazov tudi druge informacije na slikah. Potrebno pa je upoštevati, da delitev umetniških del ne more biti absolutna, saj umetnostno zgodovino sestavljajo posamezni umetniki, vsak od njih ustvarja v svojem lastnem slogu, ki lahko do neke mere sledi splošnim trendom obdobja, vendar nikoli popolnoma. Tudi posamezni likovni umetniki v času svoje kariere lahko spremenijo svoj umetniški slog.

7 ZAKLJUČEK

V članku smo pokazali nov pristop k avtomatski analizi umetniških slik z uporabo metod računalniškega vida. Demonstrirali smo, da je z metodo zaznave obrazov na slikah možno nasloviti tudi bolj kompleksna vprašanja, kot v našem primeru organizacija prostora na slikah. Čeprav rezultati te raziskave morda niso tako jasno izraženi in niso reproducirali rezultatov umetnostnih zgodovinarjev, se uporaba računalnikov na področju umetnostne zgodovine kot na sploh v humanistiki šele zares začenja. Računalniško zasnovane analitične metode bodo omogočile odgovore na vprašanja, ki si jih umetnostni zgodovinarji do sedaj sploh še niso upali zastaviti.

LITERATURA

[1] Katarina Bebar. "Upodabljanje prostora po načelih linearne perspektive s pomočjo obogatene resničnosti". V: *Likovne besede* 114 (2020), str. 14–21.

[2] Jiankang Deng in sod. "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild". V: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, str. 5202–5211. DOI: 10.1109/CVPR42600.2020.00525.

[3] David Eberly. "Least Squares Fitting of Data". V: *Magic Software, Inc.* (sep. 2001). URL: http://www.sci.utah.edu/~balling/FEtools/doc_files/LeastSquaresFitting.pdf.

[4] *Image resources: Free image resources*. Sotheby's Institute of Art. URL: https://sia.libguides.com/images/freeimageresources (pridobljeno 1. 3. 2021).

[5] Jure Kovač, Peter Peer in Franc Solina. "Automatic natural and man-made scene differentiation using perspective geometrical properties of the scenes". V: *Proceedings 15th International Conference on Systems, Signals and Image Processing*. Bratislava, 2008, 507–510.

[6] Yun Liang. *How to measure the real size of an object from an unknown picture?* Jan. 2015. URL: https://www.researchgate.net/post/How-to-measure-the-real-size-of-an-object-from-an-unknown-picture.

[7] Cory Maklin. "BIRCH Clustering Algorithm Example In Python". V: *towards data science* (jul. 2019). URL: https://towardsdatascience.com/machine-learning-birch-clustering-algorithm-clearly-explained-fb9838cbeed9.

[8] Lev Manovich. "Data Science and Digital Art History". V: *International Journal for Digital Art History* 1 (jun. 2015). DOI: 10.11588/dah.2015.1.21631.

[9] Lev Manovich. *Museum without walls, art history without names: visualization methods for Humanities and Media Studies*. Oxford Handbook Online, 2013. DOI: 10.1093/oxfordhb/9780199757640.013.005.

[10] Mohd Nayeem. "Exploring Other Face Detection Approaches(Part 1) — RetinaFace". V: *Analytics Vidhya* (jul. 2020). URL: https://medium.com/analytics-vidhya/exploring-other-face-detection-approaches-part-1-retinaface-9b00f453fd15.

[11] Robert Pepperell in Manuela Braunagel. "Do Artists Use Linear Perspective to Depict Visual Space?" V: *Perception* 43 (avg. 2014), 395 – 416. DOI: 10.1068/p7692.

[12] David G. Stork. "Computer Vision and Computer Graphics Analysis of Paintings and Drawings: An Introduction to the Literature". V: *Computer Analysis of Images and Patterns*. Ur. Xiaoyi Jiang in Nicolai Petkov. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, str. 9–24. DOI: 10.1007/978-3-642-03767-2_2.

[13] Eric W. Weisstein. "Spearman Rank Correlation Coefficient". V: *MathWorld, a Wolfram Web Resource* (brez datuma). URL: https://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html.

[14] Irving L. Zupnick. "Concept of Space and Spatial Organization in Art". V: *The Journal of Aesthetics and Art Criticism* (dec. 1959), str. 215–221. DOI: 10.2307/427268.

Automated Hate Speech Target Identification

Andraž Pelicon*
andraz.pelicon@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Blaž Škrlj*
blaz.skrlj@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Petra Kralj Novak*
petra.kralj.novak@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

We present a new human-labelled Slovenian Twitter dataset annotated for hate speech targets and attempts to automated hate speech target classification via different machine learning approaches. This work represents, to our knowledge, one of the first attempts to solve a Slovene-based text classification task with an autoML approach. Our results show that the classification task is a difficult one, both in terms of annotator agreement and in terms of classifier performance. The best performing classifier is SloBERTa-based, followed by AutoBOT-neurosymbolic-full.

KEYWORDS

hate speech targets, autoML, text features spaces

1 INTRODUCTION

Hate speech and offensive content has become pervasive in social media and has become a serious concern for government organizations, online communities, and social media platforms [13]. Due to the amount of user-generated content steadily increasing, the research community has been focusing on developing computational methods to moderate hate speech on online platforms [6, 1, 8]. While several of the proposed methods achieve good performance on distinguishing hateful and respectful content, several important challenges remain, some of them related to the data itself. Several studies report both low amounts of hate speech instances in the labelled datasets, as well as relatively low agreement scores between annotators [9]. The low agreement score between annotators indicates that recognizing hate speech is a hard task even for humans suggesting that this task requires a more broad semantic interpretation of the text and its context beyond simple pattern matching of linguistic features.

To test this assumption, we have gathered a new Slovenian dataset containing tweets annotated for hate speech targets [1]. This dataset builds on the dataset used for detecting hate speech communities [3] and topics [2] on Slovenian Twitter. The dataset is available in the clarin.si dataset repository with the handle: https://www.clarin.si/repository/xmlui/handle/11356/1398.

Next, we addressed the hate speech target classification task by the autoML approach autoBOT [10]. The key idea of autoBOT is that, instead of evolving at the learner level, evolution is conducted at the representation level. The proposed approach consists of an evolutionary algorithm that jointly optimizes various sparse representations of a given text (including word, subword, POS tag, keyword-based, knowledge graph-based and relational features) and two types of document embeddings (non-sparse representations). To our knowledge, this is one of the first attempts to solve a Slovene-based text classification task with an autoML approach. Finally, we trained a model based on the SloBERTa pre-trained language model [11], a state-of-the-art transformer-based language model pre-trained on a Slovenian corpus and a set of baselines.

Our results show that the context-aware SloBERTa model significantly outperforms all the other models. This result, together with the lower inter-annotator scores, confirms our initial assumption that hate speech target identification is a complex semantic task that requires a complex understanding of the text that goes beyond simple pattern matching. The SloBERTa model reaches annotator agreement in terms of classification accuracy, indicating a fair performance of the model.

2 DATA

We collected almost three years worth of all Slovenian Twitter data in the period from December 1, 2017, to October 1, 2020, in total 11,135,654 tweets. The period includes several government changes, elections and the first Covid-19-related lockdown. We used the TweetCat tool [5], which is developed for harvesting Twitter data of less frequent languages.

2.1 Annotation Schema

Our annotation schema is adapted from OLID [13] and FRENK [4]. It is a two-step annotation procedure. After reading a tweet, without any context, the annotator first selects the type of speech. We differentiate between the following **speech types**:

0 acceptable - non hate speech type: speech that does not contain uncivil language;
1 inappropriate - hate speech type: contains terms that are obscene, vulgar but the text is not directed at any person specifically;
2 offensive - hate speech type: including offensive generalization, contempt, dehumanization, indirect offensive remarks;
3 violent - hate speech type: author threatens, indulges, desires or calls for physical violence against a target; it also includes calling for, denying or glorifying war crimes and crimes against humanity.

If the annotator chooses either the offensive or violent hate speech type, they also include one of the twelve possible targets of hate speech:

- Racism (intolerance based on nationality, ethnicity, language, towards foreigners; and based on race, skin color)
- Migrants (intolerance of refugees or migrants, offensive generalization, call for their exclusion, restriction of rights, non-acceptance, denial of assistance …)
- Islamophobia (intolerance towards Muslims)

[*] All authors contributed equally to this research.
[1] Slovenian Twitter dataset 2018-2020 1.0: http://hdl.handle.net/11356/1423

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

- Antisemitism (intolerance of Jews; also includes conspiracy theories, Holocaust denial or glorification, offensive stereotypes . . .)
- Religion (other than above)
- Homophobia (intolerance based on sexual orientation and/or identity, calls for restrictions on the rights of LGBTQ persons
- Sexism (offensive gender-based generalization, misogynistic insults, unjustified gender discrimination)
- Ideology (intolerance based on political affiliation, political belief, ideology. . . e.g. "communists", "leftists", "home defenders", "socialists", "activists for. . . ")
- Media (journalists and media, also includes allegations of unprofessional reporting, false news, bias)
- Politics (intolerance towards individual politicians, authorities, system, political parties)
- Individual (intolerance towards any other individual due to individual characteristics; like commentator, neighbor, acquaintance)
- Other (intolerance towards members of other groups due to belonging to this group; write in the blank column on the right which group it is)

2.2 Sampling for Training and Evaluation

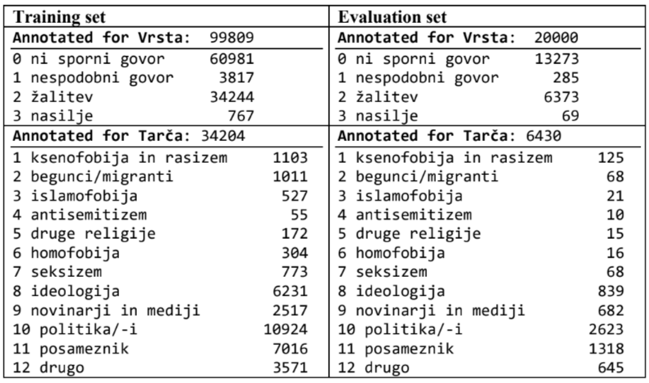
The training set is sampled from data collected before February 2020. The sampling was intentionally biased to contain as much hate speech as possible in order to obtain enough organic examples to train the model successfully. A simple model was used to flag potential hate speech content, and additionally, filtering by users and by tweet length (number of characters) was applied. 50,000[2] tweets were selected for annotation.

The evaluation set is sampled from data collected between February 2020 and August 2020. Contrary to the training set, the evaluation set is an unbiased random sample. Since the evaluation set is from a later period compared to the training set, the possibility of data linkage is minimized. Furthermore, the estimates of model performance made on the evaluation set are realistic, or even pessimistic, since the model is tested on a real-world distribution of data where hate speech is less prevalent than in the biased training set. The evaluation set is also characterized by a new topic, COVID-19; this ensures that our model is robust to small contextual shifts that may be present in the test data. For the evaluation set, 10,000 tweets were selected to be annotated.

2.3 Annotation Procedure

Each tweet was annotated twice: In 90% of the cases by two different annotators (to estimate inter-annotator agreement) and in 10% of the cases by the same annotator (to assess the self-agreement). Special attention was devoted to an evening out the overlap between annotators to get agreement estimates on equally sized sets. Ten annotators were engaged for our annotation campaign. They were given annotation guidelines, a training session and a test on a small set to evaluate their understanding of the task and their commitment before starting the annotation procedure. The annotation process lasted four months, and it required about 1,200 person-hours for the ten annotators to complete the task.

In the training set, intentionally biased in favour of hate speech, about 1% of tweets were labelled as violent, 34% as offensive (to either individuals or groups), 4% as inappropriate (mostly containing swear words), and the remaining 61% as acceptable. In the evaluation set, which is a random selection of 10.000 Slovenian tweets, only 69 tweets were labelled as violent by at least one annotator, which is about 0.3%.



Training set		Evaluation set	
Annotated for Vrsta:	99809	**Annotated for Vrsta:**	20000
0 ni sporni govor	60981	0 ni sporni govor	13273
1 nespodobni govor	3817	1 nespodobni govor	285
2 žalitev	34244	2 žalitev	6373
3 nasilje	767	3 nasilje	69
Annotated for Tarča:	34204	**Annotated for Tarča:**	6430
1 ksenofobija in rasizem	1103	1 ksenofobija in rasizem	125
2 begunci/migranti	1011	2 begunci/migranti	68
3 islamofobija	527	3 islamofobija	21
4 antisemitizem	55	4 antisemitizem	10
5 druge religije	172	5 druge religije	15
6 homofobija	304	6 homofobija	16
7 seksizem	773	7 seksizem	68
8 ideologija	6231	8 ideologija	839
9 novinarji in mediji	2517	9 novinarji in mediji	682
10 politika/-i	10924	10 politika/-i	2623
11 posameznik	7016	11 posameznik	1318
12 drugo	3571	12 drugo	645

Figure 1: Number of annotated examples for hate speech type and target. The class distribution is severely unbalanced.

The training dataset for hate speech type includes 34,204 examples and the evaluation dataset includes 6,430 examples. Many of the examples are repeated (by two annotations for the same tweet), yet conflicting (due to annotator disagreement). The training and evaluation sets for hate speech type and target are summarized in Table 1.

The overall annotator agreement for hate speech target on the training set is 63.1%, and Nominal Krippendorf Alpha is 0.537. The annotator agreement for hate speech target on the evaluation set is 62.8%, and Nominal Krippendorf Alpha is 0.503. These scores indicate that the dataset is of high quality compared to other datasets annotated for hate speech, yet the relatively low agreement indicates that the annotation task is difficult and ambiguous even for humans.

3 EXPERIMENTS

We compare different machine learning algorithms on the hate speech target identification task. They belong to one of the following three categories: classical, representation optimization and deep learning. The results are presented in Table 1.

3.1 autoBOT - an autoML for texts

With the increasing amounts of available computing power, *automation* of machine learning has become an active research endeavor. Commonly, this branch of research focuses on automatic model selection and configuration. However, it has recently also been focused on the task of obtaining a suitable representation when less-structured inputs are considered (e.g. texts). This work represents, to our knowledge, one of the first attempts to solve a Slovene-based text classification task with an existing autoML approach. The in-house developed method, called autoBOT [10], has already shown promising results on multiple shared tasks (and in extensive empirical evaluation). Albeit it commonly scores on average worse than large, multi million-parameter neural networks, it remains interpretable and does not need any specialized hardware. Thus, this system serves as an easy-to-obtain baseline which commonly performs better than *ad hoc* approaches such as, e.g. word-based features coupled

[2] Some annotators skipped some examples.

with, e.g. a Support Vector Machine (SVM). The tool has multiple configurations which determine the feature space that is being *evolved* during the search for an optimal configuration of both the representation of a given document, but also the most suitable learner. We left all settings to default, varying only the representation type, which was either symbolic, neuro-symbolic-lite, neuro-symbolic-full or neural. Detailed descriptions of these feature spaces are available online[3]. The main difference between these variants is that the neuro-symbolic ones simultaneously consider both symbolic and sub-symbolic feature spaces (e.g. tokens and embeddings of the documents), whilst symbolic or neural-only consider only one type. The neural variant is based on the two non-contextual doc2vec variants and commonly does not perform particularly well on its own.

3.2 Deep Learning

We trained a modelbased on the SloBERTa pre-trained language model [11]. SloBERTa is a transformer-based language model that shares the same architecture and training regime as the Camembert model [7] and is pre-trained on Slovenian corpora. For fine-tuning of the SloBERTa language model, we first split the original training set into training and validation folds in the 90%:10% ratio. We used the suggested hyperparameters for this model. We used the Adam optimizer with the learning rate of $2e-5$ and learning rate warmup over the first 10% of the training instances. We used a weight decay set to 0.01 for regularization. The model was trained for maximum 3 epochs with a batch size of 32. The best model was selected based on the validation set score. We performed the training of the models using the HuggingFace Transformers library [12].

We tokenized the textual input for the neural models with the language model's tokenizer. For performing matrix operations efficiently, all inputs were adjusted to the same length. After tokenizing all inputs, their maximum length was set to 256 tokens. Longer sequences were truncated, while shorter sequences were zero-padded. The fine-tuned model is available at the HuggingFace repository[4].

3.3 Other Baseline Approaches

The two mentioned approaches have demonstrated state-of-the-art performance; however, to establish their performance on this new task, we also implemented the following baselines. First, a simple majority classifier to establish the worst-case performance. Next, a doc2vec-based representation learner was coupled with a linear SVM (doc2vec). The *svm-word* is a sparse TF-IDF representation of the documents coupled with a linear SVM. Similarly, the *svm-char*, however, the representations are based on characters in this variant. The two alternatives use logistic regression (*lr-word*, *lr-char*). As another strong baseline, we used a multilingual language model called MPNet to obtain contextual representations, coupled with an SVM classifier. The baseline doc2vec model was trained for 32 epochs with eight threads. The min_count parameter was set to 2, window size to 5 and vector size to 512. For SVM and logistic regression (*LR*)-based learners, a grid search including the following regularization values was traversed: {0.1, 0.5, 1, 5, 10, 20, 50, 100, 500}.

4 RESULTS

The classification results for the discussed learning algorithms are given in Table 1. The results are sorted by learner complexity. The SloBERTa-based predictor performed the best, however, is also the one which includes the highest number of tunable parameters (more than 100m). The next series of learners are based on autoBOT's evolution and perform reasonably well. Interestingly, autoBOT variants which exploit only symbolic features perform better than the second neural network-based baseline which was not pre-trained specifically for Slovene – the *mpnet*. The remaining baselines perform worse, albeit having a similar number of final parameters to the final autoBOT-based models (tens of thousands at most). The autoBOT-neural, which implements the two main doc2vec variants, performs better than the naïve doc2vec implementation, however not notably better.

To better understand the key properties of the data set which carry information relevant for the addressed predictive task, we additionally explored *autoBOT-symbolic*'s 'report' functionality, which offers insight into the importance of individual feature subspaces. Each subspace and each feature in the subspace has a weight associated with it: the larger the weights, the more relevant a given feature type was for the learner. Visualization of these importances is shown in Table 2. It can be observed that character-based features were the most relevant for this task. This result is in alignment with many previous results on tweet classification, where e.g. punctuation-level features can be surprisingly effective. Furthermore, relational token features were also relevant. This feature type can be understood as skip-grams with dynamic distances between the two tokens. This feature type indicates that short phrases might have been of relevance. Interestingly, keyword-based features were not relevant for the learner. Further, autoBOT, being effectively a fine-tuned linear learner, also offers direct insight into fine-grained performances. Examples for the top five features per type are shown in Table 2.

5 CONCLUSION

In this work we present a new dataset of Slovenian tweets annotated for hate speech targets. To develop effective computational models to solve this task we use two approaches: the autoML approach combining symbolic and neural representations and a contextually-aware language model SloBERTa.

The results show that the context-aware SloBERTa model significantly outperforms all the other trained models. This result, together with the lower inter-annotator scores, confirm our initial assumption that hate speech target identification is a complex semantic task that requires a more complex understanding of the text that goes beyond simple pattern matching. However, the seemingly simpler models may still offer distinct advantages over the more complex neural models. First, the auto-ML models tested in this work are easily interpretable, offering insights into textual features which contribute to the classification. On the other hand, the neural language models generally work as black-boxes, and the extent of their interpretability is still an open research question. Second, the auto-ML models are significantly more straightforward to deploy as they tend to be much less computationally demanding both in terms of RAM and CPU usage. Neural language models are able to solve harder tasks but their increased number of parameters usually makes them a considerable challenge to deploy in a scalable fashion.

ACKNOWLEDGEMENTS

We would like to thank the Slovenian Research Agency for the financing of the second researcher (young researcher grant) and the financial support from research core funding no. P2-103. The

[3] autoBOT feature spaces: https://skblaz.github.io/autobot/features.html
[4] Hate speech target classification model: https://huggingface.co/IMSyPP/hate_speech_targets_slo

Table 1: Overview of the classification results. The SloBERTa model significantly outperforms all the other models and reaches inter-annotator agreement.

Classification model	Accuracy	Macro Rec	Macro Prec	Macro F1
majority	40.79%	8.33%	3.40%	4.83%
doc2vec	43.25%	20.65%	20.67%	19.76%
AutoBOT-neural (9h)	45.79%	15.37%	20.00%	16.10%
svm-word	50.39%	21.40%	25.75%	22.02%
lr-word	50.39%	21.40%	25.75%	22.02%
lr-char	51.21%	25.14%	28.17%	26.10%
svm-char	51.90%	23.47%	27.59%	24.20%
AutoBOT-neurosymbolic-lite (4h)	54.26%	27.34%	35.06%	28.90%
Paraphrase-multilingual-mpnet-base-v2 + Linear SVM	55.40%	40.24%	44.29%	41.20%
AutoBOT-symbolic (9h)	55.99%	29.68%	37.86%	31.32%
AutoBOT-neurosymbolic-full (4h)	56.28%	32.29%	37.83%	33.07%
SloBERTa	63.81%	53.03%	45.63%	48.28%

Table 2: Most relevant features per feature subspace. Feature subspaces are ordered relative to their importance. Individual numeric values next to each feature represent that feature's importance for the final learner. The features are sorted per-type. Note the word_features and their alignment with what a human would associate with hate speech.

char_features	ta s : 3.56	ni d : 2.73	lič : 2.69	ola : 2.58	ne m : 2.5
relational_features_token	pa−3−je : 2.23	pa−2−se : 2.12	v−2−pa : 1.78	ne−1−pa : 1.75	v−2−se : 1.71
pos_features	nnp nn nnp : 1.77	nnp jj nn : 1.75	nnp jj : 1.57	cc : 1.46	nn nn rb : 1.45
word_features	idioti : 1.09	riti : 0.95	tole : 0.95	sem : 0.94	fdv : 0.93
relational_features_char	e−3−d : 1.74	i−3−s : 1.56	n−3−z : 1.48	h−5−v : 1.43	z−4−t : 1.4
topic_features	topic_12 : 0.14	topic_2 : 0.02	topic_0 : 0.0	topic_1 : 0.0	topic_3 : 0.0
keyword_features	007amnesia : 0.0	15sto : 0.0	24kitchen : 0.0	2pira : 0.0	2sto7 : 0.0

work was also supported by European Union's Horizon 2020 research and innovation programme project EMBEDDIA (grant no. 825153) and the European Union's Rights, Equality and Citizenship Programme (2014-2020) project IMSyPP (grant no. 875263).[5]

REFERENCES

[1] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, 759–760.

[2] B. Evkoski, I. Mozetic, N. Ljubesic, and P. Kralj Novak. 2021. Community evolution in retweet networks. *arXiv preprint arXiv:2105.06214*.

[3] B. Evkoski, A. Pelicon, I. Mozetic, N. Ljubesic, and P. Kralj Novak. 2021. Retweet communities reveal the main sources of hate speech. (2021). arXiv: 2105.14898 [cs.SI].

[4] N. Ljubešić, D. Fišer, and T. Erjavec. 2019. The frenk datasets of socially unacceptable discourse in slovene and english. (2019). arXiv: 1906.02045 [cs.CL].

[5] N. Ljubešić, D. Fišer, and T. Erjavec. 2014. TweetCaT: a tool for building Twitter corpora of smaller languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), Reykjavik, Iceland, (May 2014).

[6] S. Malmasi and M. Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30, 2, 187–202.

[7] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, (July 2020), 7203–7219.

[8] A. Pelicon, R. Shekhar, B. Škrlj, M. Purver, and S. Pollak. 2021. Investigating cross-lingual training for offensive language detection. *PeerJ Computer Science*, 7, e559.

[9] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki. 2017. Measuring the reliability of hate speech annotations: the case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

[10] B. Škrlj, M. Martinc, N. Lavrač, and S. Pollak. 2021. Autobot: evolving neuro-symbolic representations for explainable low resource text classification. *Machine Learning*, 110, 5, 989–1028. ISSN: 1573-0565. DOI: 10.1007/s10994-021-05968-x.

[11] M. Ulčar and M. Robnik-Šikonja. 2021. Slovenian RoBERTa contextual embeddings model: SloBERTa 2.0. Slovenian language resource repository CLARIN.SI. (2021). http://hdl.handle.net/11356/1397.

[12] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.

[13] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

[5] The content of this publication represents the views of the authors only and is their sole responsibility. The European Commission does not accept any responsibility for use that may be made of the information it contains.

SiDeGame: An Online Benchmark Environment for Multi-Agent Reinforcement Learning

Jernej Puc
jernej.puc@fs.uni-lj.si
University of Ljubljana
Faculty of Mechanical Engineering
Ljubljana, Slovenia

Aleksander Sadikov
aleksander.sadikov@fri.uni-lj.si
University of Ljubljana
Faculty of Computer and Information Science
Ljubljana, Slovenia

ABSTRACT

Modern video games present a challenging benchmark for artificial intelligence research. Various technical limitations can often lead to playing interfaces that are heavily biased in terms of ease of learning for either humans or computers, and it is difficult to strike the right balance. In this paper, a new benchmark environment is presented, which emphasises the role of strategic elements by enabling more equivalent interfaces, is suitable for reinforcement learning experiments on widely distributed systems, and supports imitation learning, as is demonstrated. The environment is realised as a team-based competitive game and its source code is openly available at a public repository.

KEYWORDS

simulation environment, multi-agent system, deep neural networks, imitation learning, reinforcement learning

1 INTRODUCTION

Reinforcement learning is a powerful concept that can be used to take on highly complex challenges. In its advancement, video games have emerged as suitable benchmarks: they define clear goals, allow agents to be compared between themselves and with humans, and, in comparison to preceding milestones [7], they begin to incorporate complexities of the real world.

Success has been achieved even in notably difficult tasks, such as the modern games of StarCraft II [8] and Dota 2 [1]. However, being modern games, the authors were forced to compromise: the intricate and graphically intensive input spaces had to be simplified and transformed, while combinatorically overwhelming action spaces were functionally changed until superhuman performances could, as well, be attributed to advantages of different playing conditions.

Search for examples that could compare in strategic depth and cultivate a competitive player base, while enabling consistent interfaces and being open to researchers leaves few options but to create one anew. This has led us to create *SiDeGame*, the "simplified defusal game" (abbrev. SDG), which incorporates key rules of an established video game title in a computationally and perceptively simpler simulation environment, accessible at: `https://github.com/JernejPuc/sidegame-py`

2 RELATED WORK

Importance of an even playing field has been emphasised by authors of the For The Win (FTW) agents [4], playing a form of "capture the flag" in first-person view, while using similar input and output schemes to those of human players. However, the project is based on an inaccessible implementation of a fundamentally shallow game mode, which makes it untenable as a benchmark for reinforcement learning. Nonetheless, it shows a type of game that can suit the given requirements.

The first-person shooter (FPS) genre has many interesting representatives, some of which have already been repurposed as reinforcement learning environments [4, 5]. Unsuitably, they tend to revolve around simpler content, such as single-player or deathmatch scenarios, and are not straight-forward for researchers to customise. Indeed, accessibility and modifications generally require developer support and cooperation [6].

Confronted with this barrier, recent work on Counter-Strike: Global Offensive (CSGO) [6] resigned itself to the limits of imitation learning, which could be facilitated by external recording of public matches. Although CSGO's standard competitive mode is fittingly strategic, it, instead, focused on the mentioned deathmatch, and withheld information from agents by ignoring sound and having them use cropped and downscaled image inputs with common information omitted or rendered unrecognisable.

This paper also considers imitation learning, in attempt of establishing a baseline and starting point for eventual reinforcement learning, akin to the approach of AlphaGo [7] and AlphaStar [8]. The deep neural network architecture that was used in these experiments accepts audio inputs similarly to instances from the literature [3], which convert sounds into their frequency domain representations using the discrete Fourier transform.

3 THE SDG ENVIRONMENT

SiDeGame relies on the game rules of CSGO to provide a foundation of notable depth. Crucially, the observation space is simplified by viewing the environment from a top-down perspective in low resolution to allow modern deep neural networks to process it directly. Consequently, not all aspects of the game could be reasonably adapted and the action space could not be fully preserved, yet the playing experience remains egocentric and is largely consistent with true first-person control schemes.

3.1 Description

By the rules carried over from CSGO, two teams of 5 players each asymmetrically compete in attack and defence: the goal of one team is to detonate a bomb at one of two preset locations, while the goal of the other is to prevent them from doing so. After a certain number of rounds, the teams switch sides, and the first to pass a threshold of rounds won is declared the winner.

In the course of a round, players must navigate a map, an artificial environment with carefully placed tactical elements of various degrees of passage and cover. Besides weaponry, a player can utilise auxiliary equipment, the availability of both of which depends on prior survival and economic rewards.

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

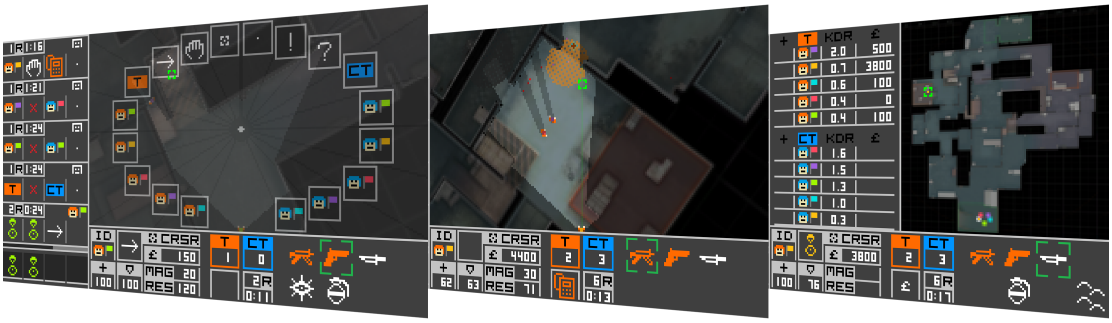


Figure 1: Screenshots of various views encountered in SiDeGame.

Additionally interesting for AI research are aspects of the game that encourage or demand active coordination, such as shared economy, unassigned roles, and imperfect information on teammates' status and surroundings.

3.2 Observations

The majority of information is provided through the image display, several screenshots of which can be seen in Figure 1. Images are generated at a low base resolution of 256×144 pixels, constraining the visual elements to be small and carefully placed, while remaining easily distinguishable. The human interface simply upscales the display with nearest neighbour interpolation, ensuring equivalence of available information.

The main view is based on projection of the radar image of a classic CSGO map, *Cache*, which has only minor vertically overlapping components and thus proved easiest to adapt. Alternative views include the inventory wheel, map plan, and communication wheels. The latter are used to construct short messages of grounded signs that are appended to the chat log in the sidebar and allow explicit coordination within the team.

Since projection is egocentric, the prominent role of sound is retained: other agents out of line of sight may still give off some information regarding their relative position, equipment, and preparedness. To support the advantages of awareness of sound, spatial audio is implemented by convolving sound signals with HRIR filters [2], while amplitude and frequency attenuation characteristics were empirically formulated. SiDeGame supports conversion of sounds into spectral vectors, which were used in the experiments of this work directly, but can also be accumulated and later processed in the form of a spectrogram.

If there is a delay between action inference and its effect in the environment, an input analogous to proprioception can also be considered. It can be trivially simulated by tracking the effective mouse and keyboard states, i. e. which keys are pressed and how the cursor is moving at a given time.

3.3 Actions

The game expects 19 binary inputs, corresponding to distinct key presses, one ternary value for scrolling the chat log, and two real values for controlling cursor movement. In general, combinations of these can legitimately be executed simultaneously, providing no benefit to the use of compound actions.

It should be noted that some of the keys, pertaining to alternative views or otherwise functional when kept held down, expect unperturbed presses lasting several seconds. For stochastic policies, where actions during training are sampled, this duration could be long enough to cause even minute probabilities to be eventually expressed, causing unintended consequences and leading to practically unplayable conditions. Training regimes should, for example, reduce the regularity of sampling, bound sampling within acceptable thresholds, or use more sophisticated contextual rules to confirm the agent's intent.

3.4 Execution

Multi-agent interaction is built upon separate server and client processes regularly exchanging state and event information via packet communication using the UDP protocol. Simulations are intended to run in real-time, but can have their tick rate and time scale adjusted on both authoritative and local ends.

With the exception of pixel-wise iteration for tracing lines of sight and disregarding the dependencies of imported extensions, the environment is fully implemented in the Python programming language. Despite clear inefficiencies, this development choice streamlines integration with machine learning solutions, which predominantly relate to the Python ecosystem, and eases code readability and customisation. Server and client processes are spawned as single Python processes that are restricted to the CPU, enabling mass parallelisation and preserving GPU resources for learning processes.

For AI agents, development targeted 30 updates per second, which had been deemed acceptable to human opponents, although higher tick rates can be achieved at both the original (144p) and reasonably upscaled (e. g. 720p) resolutions. This could also be used to speed up the simulation, subject to the computational stability and potential overhead of a specific configuration.

3.5 Online Play

In the context of agent evaluation and comparison, capability of online play, where actors, both human and artificial, can compete remotely and without having to share their program, is an essential component, as outcomes of adversarial games cannot be compared in isolation.

Feasible physical distance between actors in a match is experientially limited by temporal delays that arise from communication steps in the client loop. Inclusion of select networking concepts, such as client-side state prediction and reconciliation, foreign entity interpolation, and server-side lag compensation, should maintain playable conditions to a large extent even among international participants.

In extrapolation, online play could also support widely distributed multi-agent reinforcement learning experiments in the form of large-scale population-based training [4, 8]. These are subject to training and inference data transfer constraints, which can be alleviated by slowing down the simulation and having the data pass fewer bottlenecks. In a general configuration, multiple process groups each reserve a subset of agents (unique model parameters) from the global pool and train them with locally distributed processes, while their instances participate in shared matches, as depicted in Figure 2.

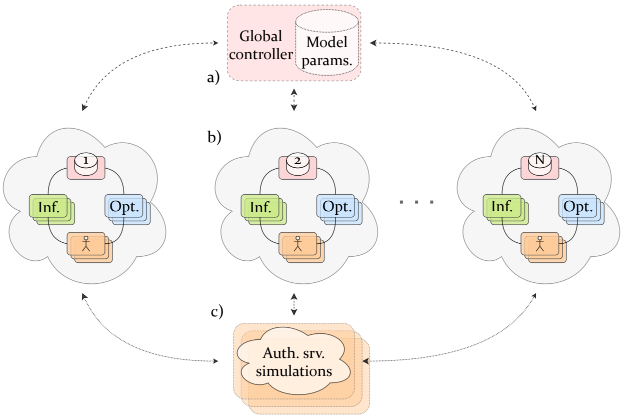


Figure 2: Online multiplayer reinforcement learning: a) The global controller process oversees all of the models in a population of agents, ensuring they are not simultaneously being updated by any two process groups. b) Process groups consist of a local controller and locally distributed inference, optimisation, and actor processes. c) All actor instances may interact through remote environments simulated by authoritative servers.

3.6 Replay System

The packets of information that a client exchanges with the server in the course of a session are made to be sufficient to faithfully reproduce the player's perspective. Byte strings can be gathered, annotated, and saved as binary files, which can then be replayed in real-time or manually stepped to inspect and extract the player's observations and actions, statistics, or other aspects of the underlying game state. Replays are an important resource for review and analysis of competitive games, but were primarily included in SiDeGame for the purposes of imitation learning.

4 SUPERVISED LEARNING BASELINE

Within the limits of available computational resources and in view of the scale of exemplary projects [1, 8], the estimated level of parallelisation, required for meaningful results of reinforcement learning experiments in an acceptable time frame, could not be reached. Instead, a baseline and a starting point for reinforcement learning was attempted to be achieved with imitation learning, a form of supervised learning from demonstrations.

4.1 Agent Model Architecture

The agent's policy was modelled as a parameterised deep neural network according to the architecture depicted in Figure 3.

The model is composed of common elements: residual convolutional blocks, recurrent cells, and fully-connected layers, forming recognisable sub-networks, such as the recurrent core, which provides the agent with memory and delay compensation, input encoding pathways, and distinct output heads.

The irregularity of visual encoding stems from the consideration that, while visual elements are simple, the display includes many of them and is relatively dense, hinting at the inevitability that not all bits of visual information can be equally accounted for at any given time. Generally, this could be addressed with sufficiently high model capacity and appropriate use of attention-based layers. In this work, however, the visual pathway was explicitly split into primary and focused visual encoding, based on the intuition of human visual perception, where only a small part of our field of view is perceived in sharp detail.

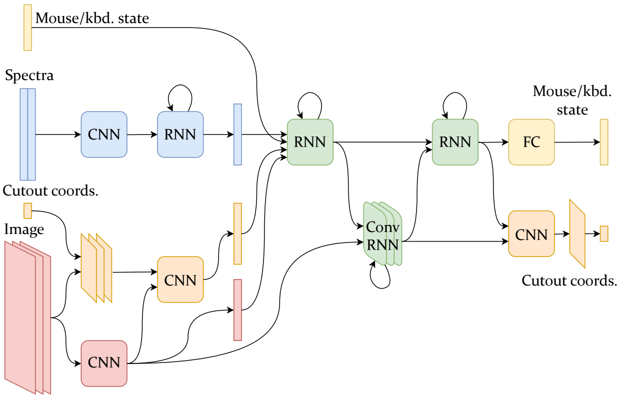


Figure 3: The deep neural network architecture used in our experiments: The visual (red), audio (blue), and mouse/keyboard state (yellow) encoding pathways converge in the recurrent core (green). Moreover, visual encoding splits off into focused encoding by cropping the input image as specified by the cutout coordinates (orange).

Instead of ingesting full-scale image data, focused visual encoding processes cutouts of much smaller size, so that singular entities can be unambiguously observed. The cutout coordinates are obtained from a spatial probability distribution along with future mouse and key states as outputs of the network. If they were, instead, determined internally, the cropping operation would need to be differentiable, which could prove hard to satisfy.

4.2 Imitation Learning

Imitation learning aims to align the agent's behaviour to that of a number of demonstrators, e. g. experienced humans. Among its basic methods is behavioural cloning, which relies on a dataset $D = \{\{o_1, a_1\}, \dots, \{o_N, a_N\}\}$ of pairs of observations o and target actions a. The agent with parameterisation θ is tasked to predict for each observation o_i such an action $\hat{a}_i$ to satisfy the following optimisation problem:

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(a_i, \hat{a}_i), \tag{1}$$

where the loss function L, evaluating similarity between predicted and imitated actions, is dependant on the form of the action space.

In this experiment, all outputs of the model were made discrete and the loss function formulated as an average of cross-entropy terms for T sub-actions of C categories:

$$L(a_i, \hat{a}_i) = -\sum_{t=1}^{T} \left(\sum_{c=1}^{C_t} a_i^{t,c} \log \hat{a}_i^{t,c} \right) \tag{2}$$

After the gradients are numerically computed with regards to the depth of truncated backpropagation through time, parameter updates are applied using one of the standard optimisation algorithms.

4.3 Demonstrations

A collection of replays was recorded from a short session between 10 demonstrators of negligible experience with SiDeGame, but with varying degrees of familiarity with related video games. Seven hours or 770,000 samples of total play were obtained at 30 frames per second, which is unideally low, especially since samples and episodes are highly correlated.

Main sub-actions were extracted from mouse and keyboard states, while focused cutout coordinates would require logistical and sensory measures that were infeasible to procure. Instead, the coordinates were manually labelled by viewing replays at 75% speed and tracing paths between estimated points of contextual interest. These labels, while not ideal, fared noticeably better than synthetically generated pseudo-labels.

Amid data extraction, observation-action pairs had actions shifted by 6 steps, conditioning the model to predict actions after a temporal delay close to the human response time.

4.4 Results

The neural network, consisting of approx. 2.9M parameters, and training procedure were implemented using the PyTorch package.

For training, a machine with 4 Nvidia 1080Ti GPUs was available. Each GPU corresponded to an optimisation process, which received an approximately equal share of training sequences and progressed them chronologically in batches of 12 sequences and epochs of 30 steps. After every epoch, the gradients with regard to the loss were computed with truncated backpropagation through time separately on each GPU, synchronously averaged between them, and used to separately update their copy of the model parameters using the AdamW optimisation algorithm with a cosine 1-cycle learning rate schedule.

The main training process ran for 300,000 steps over 6 days. The large variance in the loss in Figure 4 can be attributed to differences between game phases and subtler characteristics of demonstrators, which were found to be distinct from degrees of capability and activity.

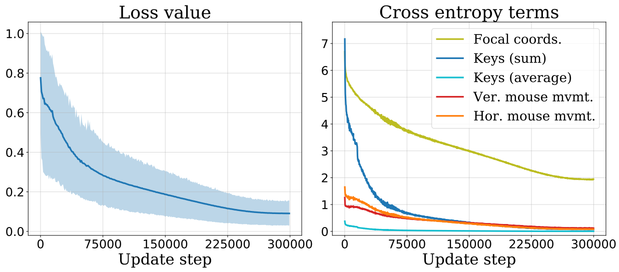


Figure 4: Loss progression over the course of training. Left: Average loss value enveloped by minimum and maximum evaluations. Right: Averages of constituent terms.

Figure 4 shows that, by the end of the training schedule, only imitation of focal coordinates leaves room for improvement, while other terms in the loss function have already overfitted. Due to the relatively small size of the network, overfitting had been underestimated, although the outcome could have been inevitable with the given amount of data.

In practice, the trained agent's behaviour was greatly sensitive to even imperceptibly slight changes in starting conditions. Its switching between alternative views was debilitatingly chaotic and had to be suppressed to allow expression of other behaviours. It seemed to respond to the presence and movement of other entities in its vicinity, was able to navigate across the map towards a tactical objective without hindering collisions and seemingly hide behind cover, but failed to demonstrate offensive behaviour.

5 CONCLUSIONS & FUTURE WORK

Attributing the shortcomings of recent works in deep reinforcement learning to inconsistencies between human and AI interfaces, a new benchmark environment has been created in the form of a lightweight multi-agent game with various tools for training and evaluation of agents. In addition to addressing these concerns, the simulation environment is based on a renowned tactical video game, providing interesting challenges for AI research, particularly in domains of sound and explicit communication.

In approaching the game with imitation learning, the trained agent failed to develop practically meaningful behaviours when trained on arguably few demonstrations and was found lacking as a starting point for reinforcement learning experiments. Nevertheless, the presented agent model architecture is general enough to be applicable to other common tasks with standard computer peripherals and lends itself to further experimentation.

Online characteristics of the created environment hint at its potential for large-scale reinforcement learning experiments, with its accessibility and adaptability allowing the AI community to explore this and other directions. At the same time, certain components of the environment that are not specific to AI research could also prove useful to a wider community, outside of the scope of its primary intent.

REFERENCES

[1] Christopher Berner et al. 2019. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680. arXiv: 1912.06680. http://arxiv.org/abs/1912.06680.

[2] Fabian Brinkmann et al. 2017. A high resolution and full-spherical head-related transfer function database for different head-above-torso orientations. *J. Audio Eng. Soc*, 65, 10, 841–848. DOI: 10.17743/jaes.2017.0033. http://www.aes.org/e-lib/browse.cfm?elib=19357.

[3] Shashank Hegde, Anssi Kanervisto, and Aleksei Petrenko. 2021. Agents that listen: high-throughput reinforcement learning with multiple sensory systems. *CoRR*, abs/2107.02195. arXiv: 2107.02195. https://arxiv.org/abs/2107.02195.

[4] Max Jaderberg et al. 2018. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. *CoRR*, abs/1807.01281. arXiv: 1807.01281. http://arxiv.org/abs/1807.01281.

[5] Michal Kempka et al. 2016. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. *CoRR*, abs/1605.02097. arXiv: 1605.02097. http://arxiv.org/abs/1605.02097.

[6] Tim Pearce and Jun Zhu. 2021. Counter-Strike deathmatch with large-scale behavioural cloning. *CoRR*, abs/2104.04258. arXiv: 2104.04258. https://arxiv.org/abs/2104.04258.

[7] David Silver et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529, 7587, 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961. https://doi.org/10.1038/nature16961.

[8] Oriol Vinyals et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575, 7782, 350–354. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1724-z. https://doi.org/10.1038/s41586-019-1724-z.

Question Ranking for Food Frequency Questionnaires

Nina Reščič
nina.rescic@ijs.si
Department of Intelligent Systems, Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Mitja Luštrek
mitja.lustrek@ijs.si
Department of Intelligent Systems, Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Food Frequency Questionnaires (FFQs) are probably the most commonly used dietary assessment tools. In the WellCo project, we developed the Extended Short Form Food Frequency Questionnaire (ESFFFQ), integrated into a mobile application, in order to monitor the quality of users' nutrition. The developed questionnaire returns diet quality scores for eight targets — *fruit intake*, *vegetable intake*, *fish intake*, *salt intake*, *sugar intake*, *fat intake*, *fibre intake* and *protein intake*. This paper explores the single-target problem of question ranking. We compared the question ranking of the machine learning algorithms on three different types of features for classification and regression problems. Our findings showed that the addressing problem as a regression problem performs better than treating it as a classification problem and the best performance was achieved by using a Linear Regression on features, where answers were transformed to frequencies of consumption of certain food groups.

KEYWORDS

nutrition monitoring, FFQs, question ranking

1 INTRODUCTION

Adopting and maintaining a healthy lifestyle has become extremely important and healthy nutrition habits represent a major part in achieving this goal. Self-assessment tools are playing a big role in nutrition monitoring and many applications are including Food Frequency Questionnaires (FFQs) as a monitoring tool, due to they in-expensiveness, simplicity and reasonably good assessment [8, 3]. An FFQ is a questionnaire that asks the respondents about the frequency of consumption of different food items (e.g., "How many times a week do you eat fish?"). In the EU-funded project WellCo we developed and validated an Extended Short Form Frequency questionnaire (ESFFFQ) [5] that was included in a health coaching application for seniors.

Cade et al. [2] suggest that for assessment of dietary data short FFQs could be sufficient and that marginal gain in information is decreasing with extensive FFQs. Block et al. [1] concluded that longer and reduced return comparable values of micronutrients intake. Taking this idea a step forward, we explored the possibilities to get the most information even if one does not answer the whole questionnaire. In our previous work we explored how to find the smallest set of questions that still provides enough information by applying different feature selection techniques [6, 7].

This paper explores the ranking of questions and is the next step from our previous work. With ranking the questions by importance and asking them in the ranked order, it can be expected that quality of predictions will improve with each additional answer and we are not limited with the constraint that certain number of questions should be answered. We addressed the problem as a single-target problem for classification and regression. Additionally, we tested the algorithms on different representations of features for both type of problem. The findings of this paper could be used for setting the baseline for our future research.

2 METHODOLOGY

2.1 Problem outline

In our previous research [6, 7] we tried to find subsets of questions that would allow us to ask the users about their dietary habits with as few questions as possible and still get sufficient information to evaluate their nutrition. For this we used the Extended Short Form Food Frequency Questionnaire (ESFFFQ) [5]. The questionnaire returns diet quality scores scores for *fruit intake*, *vegetable intake*, *fish intake*, *salt intake*, *sugar intake,fat intake*, *fibre intake* and *protein intake*. We calculate the nutrient intake amounts and from there we further calculate the diet quality scores.

The questionnaire was included in a mobile application, where the system asked the users about their diet with one or two questions per day. The answers were saved into a database and every fortnight the quality scores were recalculated. As it could happen that the users did not answer all the questions by the time the recalculation was done, it was of great importance to ask the questions in the right order. In the terminology of machine learning this would be a feature ranking problem. We explored the problem as a set of single-target problems — separately for individual outcome scores. As three of the diet quality scores (*fruit*, *vegetable* and *fish intake*) are only dependent on one or two questions, the problem of feature ranking is trivial. Therefore we explored the problem for the remaining five targets — *fat intake*, *sugar intake*, *fibre intake*, *protein intake* and *salt intake*.

2.2 Dataset

We got the answers to ESFFFQ from 92 adults as a part of the WellCo project and additionally from 1039 adults included in SIMenu, the Slovenian EUMenu research project [4]. The questions included in the ESFFFQ were a subset of the questions in the FFQ in SIMenu. Furthermore, the answers (consumption frequencies) were equivalent in both questionnaires, and consequently extracting the answers from SIMenu and adding them to the answers from the ESFFFQ was a very straightforward task.

2.3 Feature ranking

To do the experiments, we first randomly split the data into validation and training sets in ratio 1:3. To train the models and

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

rank the features we then used 4-fold cross-validation on the training set and used the average feature importance from all 4 folds as the final feature ranking.

The ranked features were used to predict quality scores (classification problem) and nutrient amount (regression problem), by adding the question as they were ranked. In this paper we present the results for two commonly used machine learning algorithms — Logistic/Linear Regression and Random Forest Classifier/Regressor. To rank the features we used the absolute value of the coefficients in the Linear/Logistic Regression and the `feature_importance` attribute as implemented in the Random Forest Classifier/Regressor in the `sklearn` library.

Additionally we compared different feature representations — features where answers are represented with nominal discrete equidistant values (once per week is represented as integer 2), features where answers were transformed into frequencies of consumption (once per week is represented as approx. 0.14 per day) and features where answers were transformed into amounts of nutrients (once per week is represented as grams/day). In the last represenation, the features differed between the targets *sugar*, *fat*, *salt*, *fibre* and *protein*. We ran the experiments for five diet categories (*fat intake*, *sugar intake*, *fibre intake*, *protein intake* and *salt intake*) for both classification and regression problem. In both cases we started with the best ranked question, trained the model and compared results on train and validation sets. Then we added the second best ranked question, trained the models and compared the results. We added the questions one by one until the last one.

3 RESULTS

3.1 Classification problem

For classification we tried to predict the quality scores for each of the five nutrition categories. There were three scores - 2 (good), 1 (medium) and 0 (bad). The distribution of the scores for all the categories is shown in Table 1.

Table 1: Distribution of target values for classification

Score	Fat	Sugar	Fibre	Protein	Salt
2	51%	74%	26%	79%	32%
1	31%	14%	22%	13%	47%
0	18%	12%	52%	8%	21%

We compared Random Forest Classifier and Logistic Regression for three different types of features - discrete equidistant answers, answers transformed to frequencies and answers transformed to amounts.

Fat. For Random Forest (RF) there was not a big difference between the three representations of the features. With all three, the highest accuracy on the validation set (79%) is achieved with 5 questions and afterwards the accuracy starts falling and stays on the interval between 75% and 79%. This clearly indicates overfitting, which is confirmed by the fact that the accuracy for RF on the training set was 100% from the fifth question. A similar situation happened for all the remaining targets and will not be repeated in the following subsections. On the training set Logistic Regression (LR) had worse results than the RF and it also performed the worst from all algorithms when run on the discrete features. However, when the features are transformed into frequencies or amounts, we get better results on the validations set than with RF.

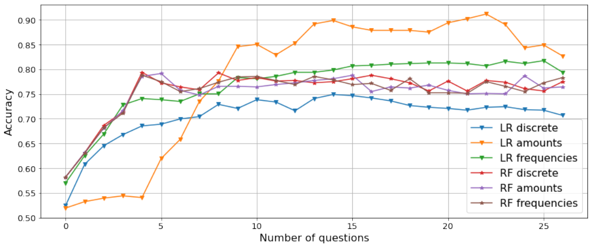


Figure 1: Results on validation set for *fat intake*

Sugar. For *sugar intake* the story is very similar. RF performed fairly well for the first few questions and then the accuracy began to fall. The best performing algorithm was the LR on the features (Figure 2, where the answers were transformed into frequencies.

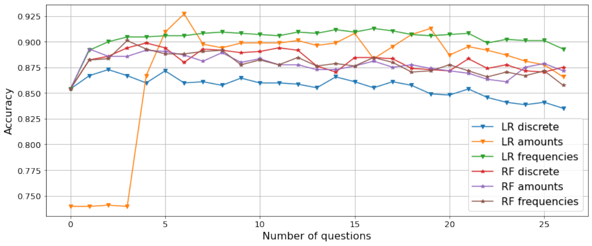


Figure 2: Results on validation set for *sugar intake*

Fibre. For *fibre intake* the RF algorithms performed better for a very long time (Figure 3) and it reached the best accuracy after 6 questions. The LR performed worse, and it did similarly badly on the training set as well.

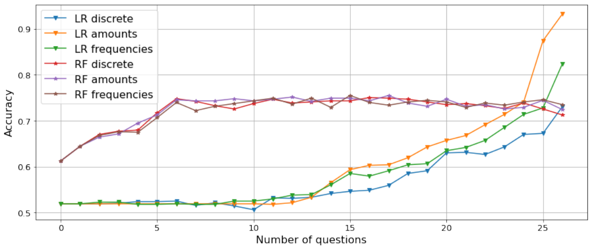


Figure 3: Results on validation set for *fibre intake*

Protein. For *protein intake* (Figure 4) the results are similar to those for *fibre intake*. However, in case of *protein intake* the majority class is 79% and most of the algorithms almost never exceeded this value.

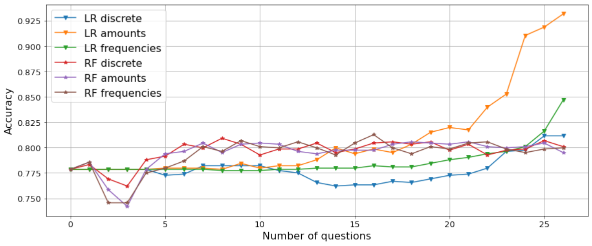


Figure 4: Results on validation set for *protein intake*

Salt. For *salt intake* the best model is the LR on the answers transformed to amounts. As seen in Figure 5, it exceeded the RF algorithms for almost 20% from eleventh added question on and predicted the quality scores with more than 90% accuracy with only 14 questions, which is half of the questionnaire.

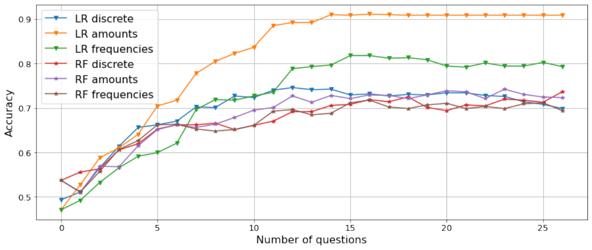


Figure 5: Results on validation set for *salt intake*

3.2 Regression problem

While knowing the quality score is a valid first information whether one's diet is good or not, generally more interesting information is how good (or how bad) it really is. Therefore it is reasonable to look at the same problem as a regression problem , where we try to predict the actual amount (in grams) of consumed nutrients. Again we explored the performance of Random Forest Regressor (RF) and Linear Regression (LR) on the three previously described feature sets.

Table 2: Nutrient intake in grams/day to quality scores

Score	Fat[g]	Sugar[g]	Fibre[g]	Protein[g]	Salt[g]
2	≤ 74	≤ 55	≥ 30	≥ 55	≤ 6
1	else	else	else	else	else
0	≥ 111	≥ 82	≤ 25	≤ 45	≥ 9

Fat. The best performing algorithm for *fat intake* was the LR on the answers transformed to frequencies. The overfitting of the RF is even more visible than with the classification problem as the errors for these models did not fall under 20 grams even if all the questions were used, while the error of the LR on the feature sets where the answers are transformed to frequencies or amounts was smaller than 5 grams from eleven included questions (Figure 6).

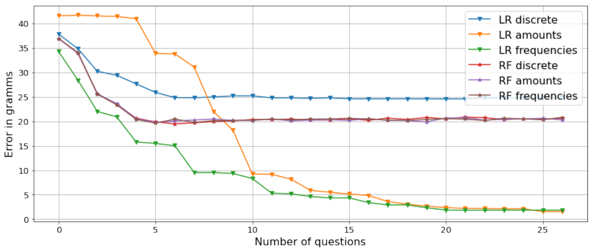


Figure 6: Results on validation set for *fat intake*

Sugar. Similarly to *fat intake*, LR with the 'frequency features' performed best (Figure 7). However the LR on the 'amounts features' performed well for more than 15 questions, but predicted the worst for the first eleven included questions.

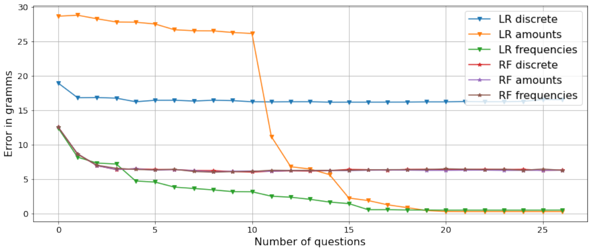


Figure 7: Results on validation set for *sugar intake*

Fibre. Classification for *fibre intake* was very bad, however, when considering it as a regression problem, the LR on 'frequency' features' predicted the amounts with error smaller than 2 grams when more than eleven questions were used 8. Considering Table 2 this means that predicting how bad/good the *fibre intake* was done better then predicting if it is bad or good.

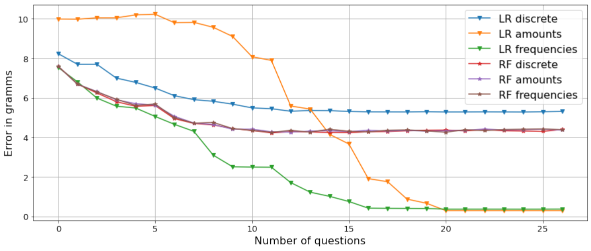


Figure 8: Results on validation set for *fibre intake*

Protein. For *protein intake* all algorithms had a similar performance up to ten included questions, however, the LR on the 'frequency features' started to perform better and better with each added questions and predicted the amount of protein consumption with error of 5 grams (Figure 9).

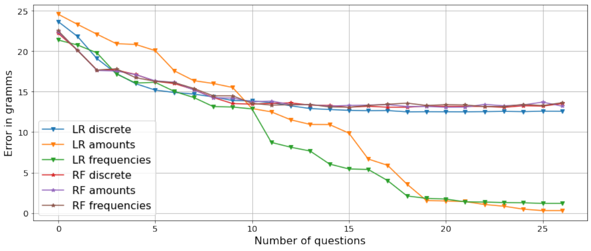


Figure 9: Results on validation set for *protein intake*

Salt. Similarly to the *protein intake* all algorithms performed with a comparable error up to nine included questions, and after that LR using the features transformed to frequencies started to perform way better and predicted *salt intake* with error smaller than 1 gram with eleven included questions (Figure 10).

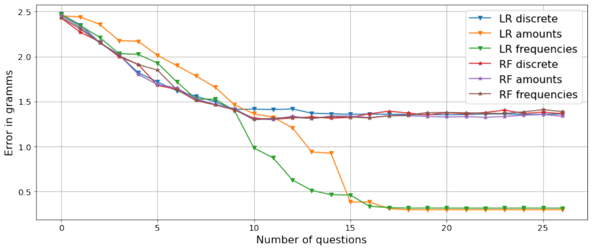


Figure 10: Results on validation set for *salt intake*

3.3 Discussion

We compared performance of feature ranking for two different machine learning algorithms on three different types of features for both classification and regression problems. While the classification problem might give the general idea about one's dietary habits, it is inclined towards overfitting even for very simple models, such as Logistic Regression, while more complex algorithms, Random Forest Classifier in our case, are even more subject to this deficiency. By predicting amounts instead of quality scores, one gets information about how good/bad the dietary habits are instead of just if they are good or bad.

Transforming features from discrete equidistant values to frequencies or amounts of nutrients proved to be a very good approach. The transformation gave better results for both classification and regression problem for both Random Forest Regressor/Classifier and Logistic/Linear Regression. While the performance of both algorithms on features transformed to frequencies and features transformed to amounts for the classification problem was comparable, and Linear Regression on features transformed to amounts gave markedly better results for *salt intake*, the Linear Regression on features transformed to frequencies outperformed all other combinations of features and algorithms for the regression problem for all of the targets. The reason for this is that linear regression on amounts is a very good match in the sense that the target variable (total amount) is the sum of all features (partial amounts).

Transforming the features to frequencies instead to amounts has another advantage — frequencies transformed to amounts are specific to each target, while features transformed to frequencies are equal for all targets. This is an important finding for possible future research where one would address ranking of questions as a multi-target problem. Additionally, regression problem using Linear Regression on features transformed to frequencies could solve as a baseline for future experiments.

4 CONCLUSION AND FUTURE WORK

Ranking the questions of FFQs when it could be expected that not all of the questions will be answered is an important step when building models for predicting quality of one's diet. In this paper we compared two feature ranking algorithms on three different types of features for classification and regression problem for five targets. The findings of this paper show that considering the problem as a regression problem on features transformed to frequencies and using a simple machine learning algorithms (Linear Regression) gives the best results for all five targets and provides baseline for future experiments.

There are several possibilities for future work. As hinted in the previous section, the question of multi-target question ranking is one of the first that appears — one might want to monitor several nutrition quality scores but still would want to avoid answering too many questions. Next, probably more important and interesting research problem, is how to use the answers already provided to our advantage — so instead of statically ranking the questions we would rather explore how we could improve the prediction performance by dynamically ranking and asking the questions.

ACKNOWLEDGMENTS

WellCo Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769765.

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0209).

The WideHealth project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 95227.

REFERENCES

[1] Block G, Hartman AM, and Naughton D. 1990. A reduced dietary questionnaire: development and validation. *Epidemiology*, 1, 58–64. DOI: 10.1097/00001648-199001000-00013.

[2] Cade J., Thompson R., Burley V., and Warm D. 2002. Development, validation and utilisation of food-frequency questionnaires – a review. *Public Health Nutrition*, 5, 4, 567–587. DOI: 10.1079/PHN2001318.

[3] Shim JS, Oh K, and Kim HC. 2014. Dietary assessment methods in epidemiologic studies. *Epidemiol Health*, 36. DOI: 10.4178/epih/e2014009.

[4] Gregorič M, Blaznik U., Delfar N., Zaletel M., Lavtar D., Koroušić-Seljak B., Golja P., Zdešar Kotnik K., Pravst I., Fidler Mis N., Kostanjevec S., Pajnkihar M., Poklar Vatovec T., and Hočevar-Grom A. 2019. Slovenian national food consumption survey in adolescents, adults and elderly : external scientific report. *EFSA Supporting Publications*, 16, 11, 1729E. DOI: 10.2903/sp.efsa.2019.EN-1729.

[5] Reščič N., Valenčič E., Mlinarič E., Seljak Koroušić B., and Luštrek M. 2019. Mobile nutrition monitoring for well-being. In (UbiComp/ISWC '19 Adjunct). Association for Computing Machinery, London, United Kingdom, 1194–1197. DOI: 10.1145/3341162.3347076.

[6] Reščič N., Eftimov T., Koroušić Seljak B., and Luštrek M. 2020. Optimising an ffq using a machine learning pipeline to teach an efficient nutrient intake predictive model. *Nutrients*, 12, 12. DOI: 10.3390/nu12123789.

[7] Reščič N., Eftimov T., and Seljak Koroušić B. 2020. Comparison of feature selection algorithms for minimization of target specific ffqs. In *2020 IEEE International Conference on Big Data (Big Data)*, 3592–3595. DOI: 10.1109/BigData50022.2020.9378246.

[8] Thompson T. and Byers T. 1994. Dietary assessment resource manual. *The Journal of nutrition*, 124, (December 1994), 2245S–2317S. DOI: 10.1093/jn/124.suppl_11.2245s.

Daily Covid-19 Deaths Prediction For Slovenia

David Susič
"Jožef Stefan" Institute
Ljubljana, Slovenia
david.susic@ijs.si

ABSTRACT

In this paper, models for predicting daily Covid-19 deaths for Slovenia are analysed. Two different approaches are considered. In the first approach, the models were trained on the fist wave dataset of state intervention plans, cases and country-specific static data for 11 other European countries. The models with the best performance in this case were the k-Nearest Neighbors regressor and the Random Forest regressor. In the second approach, a time-series analysis was performed. The models used in this case were Seasonal Autoregressive Integrated Moving Average Exogenous and Feed forward Neural Network. For comparison, all 4 models were tested on the second wave for Slovenia and the model with the best performance was Feed forward Neural Network, with a mean absolute error of 1.34 deaths.

KEYWORDS

Covid-19, deaths, predictions, machine learning

1 INTRODUCTION

The aim of this analysis is to find out whether we can predict Covid-19 deaths for Slovenia based on the characteristics of the epidemic in other European countries, and whether we can predict deaths based on a time series analysis of historical data (e.g. predicting for the second wave based on the first wave information). The main advantage of the first approach is that we do not need historical case and death data for the country for which we are making a prediction (in this case Slovenia), while the second approach is generally more accurate but relies on historical death data. The aim is also to find out which of the two approaches provides more accurate predictions. It is important to note that although this is a study for Slovenia, the results can be interpreted as a general assessment of the effectiveness of the methods described for predicting Covid-19 deaths and can be applied to any country for which the data are available.

The data used in this analysis are described in Section 2. Section 3 provides a description of the approaches and the models. Section 4 contains a discussion of the determination of the optimal parameters of the selected models. The results are given in Section 5. The conclusion, along with ideas for possible improvements, is given in Section 6.

2 DATA DESCRIPTION AND PREPARATION

The data used in this paper consist of daily Covid-19 related features at the country level. It contains 12 different Covid-19 related government interventions (school closing, workplace closing, cancel public events, restrictions on gatherings, close public transport, stay at home requirements, restrictions on internal movement, international travel controls, public information campaigns, testing policy, contact tracing, and facial coverings), Covid-19 related cases and deaths, and some static data, in particular the country's population, population density, median age, percentage of people over 65, percentage of people over 70, gdp per capita, cardiovascular death rate, diabetes prevalence, percentage of female and male smokers, hospital beds per thousand people, and life expectancy. To suppress anomalies in registered cases on Sundays and holidays, a 7-day moving average was used for both cases and deaths. The dataset covers the European countries of Slovenia, Italy, Hungary, Austria, Croatia, France, Germany, Poland, Slovak Republic, Bosnia and Herzegovina, and the Netherlands from January 22, 2020 to December 11, 2020. All of the countries chosen for this study are geographically next to one another and are thus expected to have similar course of epidemic. The data on government interventions, cases and deaths are derived from the "COVID-19 Government response tracker" database, collected by Blavatnik School of Government at Oxford University [4]. The intervention values range between 0-4 and represent their strictness, for example, if only some or all schools are closed. The static data are collected from a variety of sources (United Nations, World Bank, Global Burden of Disease, Blavatnik School of Government, etc.) [3]. The original data are publicly available online. The processed data used for the purpose of this study can be found online at https://repo.ijs.si/davidsusic/covid-seminar-data.

3 METHODS AND MODELS

Two different approaches were considered for the analysis. For the first part of the analysis, referred to as the country-specific approach, the models were trained on the data of government intervention plans, cases, deaths and country-specific static data for the 10 other European countries, with the aim of predicting deaths for Slovenia. In this case, the predictions were made for each day, disregarding the time order. For the second part of the analysis, a time series prediction was performed, using only the daily deaths for Slovenia as data.

3.1 Country-Specific Approach

In the country-specific approach, the selection of the base model was very important, as models that perform worse than the base model are not worthy of interpretation. The baseline was defined as

$$N_{\text{deaths}}(t) = N_{\text{cases}}(t-14) \cdot M, \qquad (1)$$

where $M = 0.023$ is the mortality rate factor of those infected, calculated as a weighted average of the mortality rates of the countries included in this study [2], and t denotes a specific day. This simple model implies, that the number of deaths on a given day t is equal to the number of new infections on the

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

day $t - 14$, multiplied by the mortality rate factor. The regressor model that were tested are: Random Forest (RF), k-Nearest Neighbors (KNN), Stochastic Gradient Descent, Ridge, Lasso, and Epsilon-Support Vector. Description of all of the models can be found in the Python scikit-learn documentation [5]. The two that performed significantly better than the baseline were the KNN regressor and RF regressor. Other regression models performed the same or worse than the baseline model and were thus not used in the further analysis. All models were tested in the 10-fold cross-validation with the performance measures mean absolute error (MAE), mean squared error (MSE) and R^2 score on the data subset that does not include Slovenia. The measures are defined as:

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i|, \tag{2a}$$

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2, \tag{2b}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}, \tag{2c}$$

where $\hat{y}$ is the predicted value of the i-th sample, y_i is the corresponding true value, n is the sample size and $\bar{y}$ is the average true value $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_1$.

For each sample, additional features of the government interventions and cases were added for the previous days. The number of previous days was defined using the lookback parameter. Models were tested for lookback values between −28 and 0 days. The comparison is shown in Figure 1. It can be seen that the performance decreases in the range where the lookback is shorter than 14 days, but does not increase in the range where the lookback exceeds this value. The main reason for this is probably the fact that most deaths occur within the first 14 days of infection. A lookback of 14 days was used for further analysis as it was found to be the most appropriate.

3.2 Time-Series Approach

In the second approach, a time series analysis was performed. In this case, only daily deaths for Slovenia were used as data. The models used in this case were Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX(p,d,q)(P,D,Q,m) [6] and Feed forward Neural Network (FFNN) [1].

The former is a combination of several different algorithms. The first is the autoregressive AR (p) model, which is a linear model that relies only on past p values to predict current values. The next is the moving average MA (q) model, which uses the residuals of the past q values to fit the model accordingly. The I(d) represents the order of integration. It represents the number of times we need to integrate the time series to ensure stationarity. The X stands for exogenous variable, i.e., it suggests adding a separate other external variable to measure the target variable. Finally, the S stands for seasonal, meaning that we expect our data to have a seasonal aspect. The parameters P, D, and Q are the seasonal versions of the parameters p, d, and q, and the parameter m represents the length of the cycle.

The FFNN structure included 10 input perceptrons - one for each death value in the last 10 days, a hidden layer of 64 perceptrons, and 1 output perceptron.

Since the future data of the time series contain the information about the past, a forward chaining approach was performed for n-fold cross validation. This means, that there is no random shuffling of the data. The test set must always be the final portion of the data - the final part of the date range. The concept of forward chaining is shown in Figure 2. The results of the 10-fold cross-validation of the predictions for 21 days are shown in Table 1.

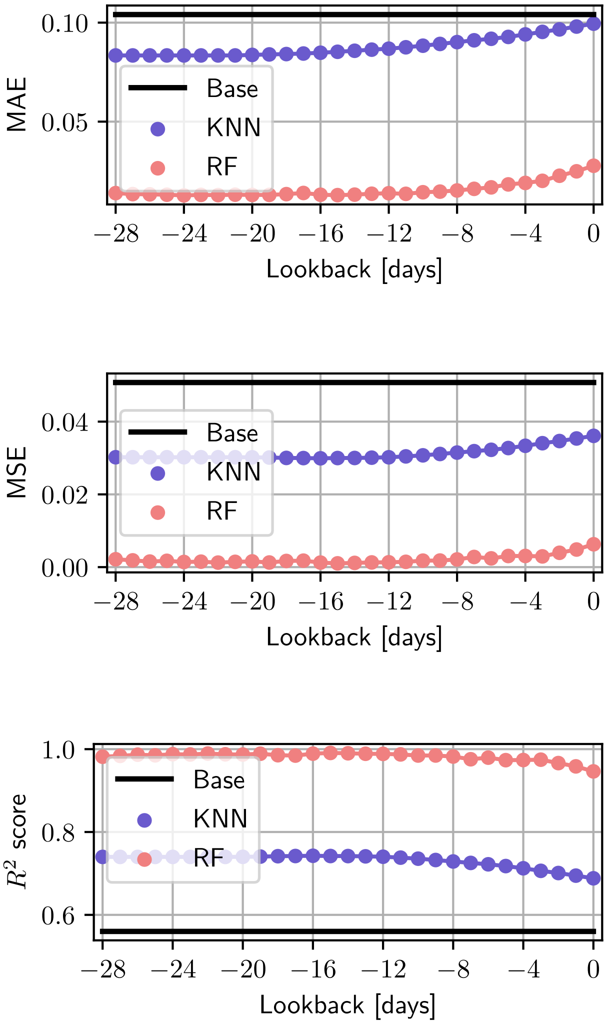


Figure 1: 10-fold cross validation performance measure of the models for different lookback parameter. The measures and its units are are: MAE [deaths/100k] (top), MSE [deaths^2/100k^2] (middle) and R^2 score (bottom)

Table 1: 10-fold cross-validation performance measures of the predictions for 21 days for SARIMAX and FFNN algorithms.

	MAE [deaths]	MSE [deaths^2]	R^2 score
SARIMAX	1.13	4.81	0.71s
FFNN	0.53	1.15	0.88

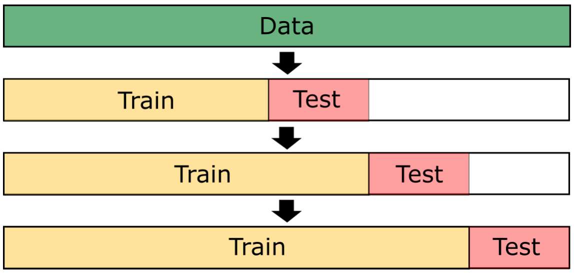


Figure 2: Forward chaining approach to time-series n-fold cross-validation.

4 MODELS' PARAMETERS SELECTION

The next step was to determine the optimal parameters of the selected models. For this purpose, the regressor models were trained on the same dataset used in the 10-fold cross-validation and tested on the data for Slovenia. For this particular case, different model parameters were tested to see which performed best. The MAE [deaths/100k] as a function of parameters K for the KNN and as a function of the number of trees for RF are shown in the Figures 3 and 4, respectively.

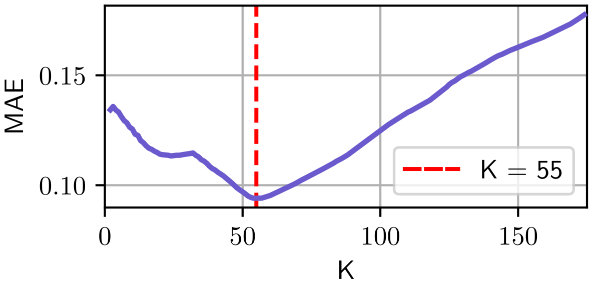


Figure 3: MAE of the KNN regressor as function of K.

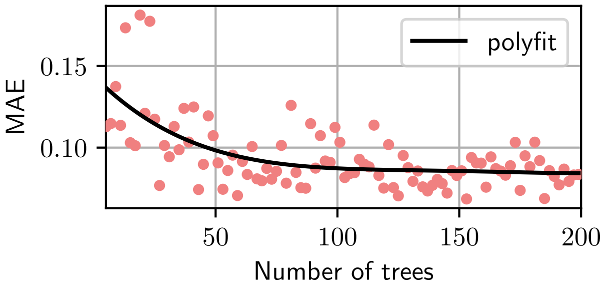


Figure 4: MAE of the RF regressor as a function of the number of trees.

For the KNN regressor, MAE has a minimum at $K = 55$, while for RF the fitting function shows that the appropriate number of trees is 100, since the model does not improve with additional trees at this point. It is important to note that since RF is random in the sense that it randomly selects a subset of features at each splitting decision, the results and hence the performance measures are also somewhat random. However, they do follow a certain trend that becomes apparent when a polyfit is applied. To reduce the randomness of the results, the average of 3 separate predictions was calculated for each number of trees.

To determine the best parameters of the SARIMAX model, the *auto_arima* algorithm from the Python *pmdarima* library was used [7]. The algorithm analyzes the given data and determines the best model and its parameters for that data. In this case, the selected model was SARIMAX(2, 1, 4)(4, 1, 1, 12).

In the case of FFNN, the parameter selection was omitted - the same model structure was always used.

5 RESULTS

With the optimal parameters selected, the graphs of the predictions can be plotted. The predictions of the country-specific approach are shown in Figure 5.

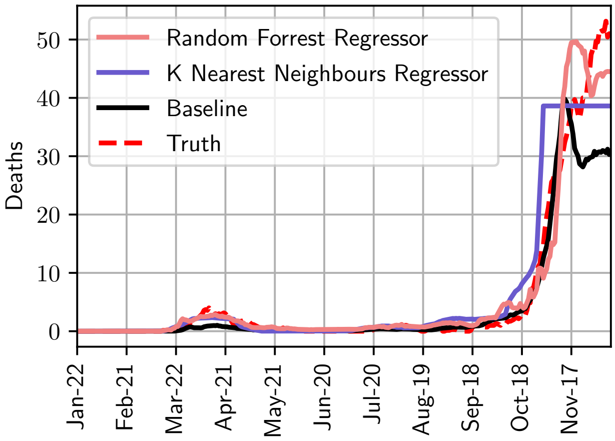


Figure 5: Deaths for Slovenia from 22.1.2020 to 11.12.2020. Models' predictions, compared to true values.

All models predicted the number of deaths for the first epidemic wave fairly accurately. As a result of the unrepresentative reporting of Covid-19 cases for the second wave, the base model predicts a much lower number of daily deaths. We can also see that the KNN regressor predicts the same value from a certain day forward. The reason for this is most probably that the algorithm always finds the same k=55 neighbors, thus always predicts the same value. To avoid this, a larger dataset would be required. MAE for RF, KNN and baseline are shown in Table 2.

Table 2: MAE comparison of the country-specific models for the interval from 22.1.2020 to 11.12.2020.

	RF	KNN	baseline
MAE [deaths]	5.41	5.39	5.48

The predictions for the time interval between 21.11.2020 and 11.12.2020 for the time-series approach are shown in Figure 6. MAE for FFNN and SARIMAX, shown in Table 3, are substantionally lower than MAE of the country-specific models. However, the accuracy decreases as the prediction time interval increases.

Table 3: MAE comparison of the time-series models for the interval from 21.11.2020 to 11.12.2020.

	FFNN	SARIMAX
MAE [deaths]	1.24	2.27

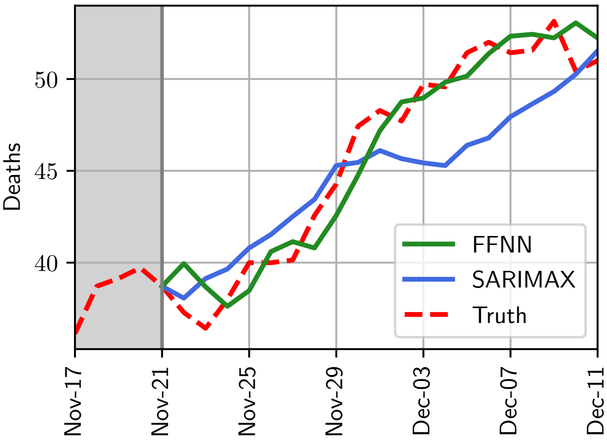


Figure 6: Slovenia deaths from 21.11.2020 to 11.12.2020. Time-series models' predictions, compared to true values.

To determine the overall best model for such predictions, all 4 models were tested on the second epidemic wave. The predictions are visualized in the Figure and the MAEs [deaths] are listed in the Table 4.

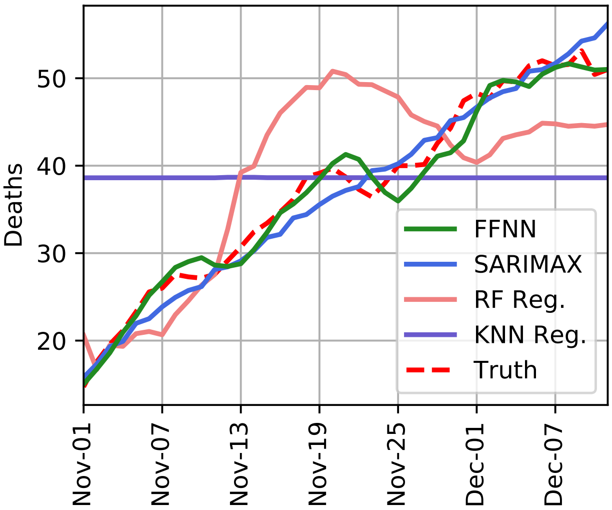


Figure 7: Slovenia deaths from 1.11.2020 to 11.12.2020. Models' predictions, compared to true values.

Table 4: MAE comparison of the models for the interval from 1.11.2020 to 11.12.2020.

	FFNN	SARIMAX	RF Reg.	KNN Reg.
MAE [deaths]	1.34	1.67	6.46	8.85

It can be seen that in this case the time-series approach is more accurate than the country-specific one. However, for longer time intervals, the country-specific approach is better because it does not rely on past data. It is important to note that the country-specific models' error are actually lower when making predictions from the start of the epidemic. The reason for this is that for the first 6 months, the numbers of deaths were very low as can be seen in the Figure 5.

The best performing model overall is the FFNN with the MAE of 1.34 deaths. The reason for the best performance of this model is probably that it had a relatively high number of input parameters. The input layer consisted of 10 perceptrons, i.e. each prediction was based on the values of the last 10 days.

6 CONCLUSION

In this paper, two different approaches to predicting Covid-19 deaths for Slovenia were tested. Both approaches turned out to be reliable. The main implications of the presented study are that for short time intervals the time series approach is much more accurate than the country-specific approach. The advantage of the country-specific approach is that it can predict the number of deaths for a given day, based on the number of cases, countermeasures and country-specific static data, without necessarily having information about the past. On the other hand, for the prediction of the second wave, where we already know the course of the epidemic in the first wave, the time series approach is better - at least for the prediction for Slovenia. In the future studies, predictions for the third and fourth waves will be analysed.

REFERENCES

[1] Francois Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

[2] Ensheng Dong et al. 2020. An interactive web-based dashboard to track covid-19 in real time. *The Lancet Infectious Diseases*, 20, 5. DOI: 10.1016/S1473-3099(20)30120-1. http://doi.acm.org/10.1016/S1473-3099(20)30120-1.

[3] Thomas Hale et al. 2020. A cross-country database of covid-19 testing. *Scientific Data*, 7, 345. DOI: 10.1038/s41597-020-00688-8. http://doi.acm.org/10.1038/s41597-020-00688-8.

[4] Thomas Hale et al. 2021. A global panel database of pandemic policies (oxford covid-19 government response tracker). *Nature Human Behaviour*, 5, 3529–538. DOI: 10.1038/s41562-021-01079-8. http://doi.acm.org/10.1038/s41562-021-01079-8.

[5] Fabian Pedregosa et al. 2012. Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12, (January 2012).

[6] Skipper Seabold and Josef Perktold. 2010. Statsmodels: econometric and statistical modeling with python. *Proceedings of the 9th Python in Science Conference*, 2010, (January 2010).

[7] Taylor G. Smith et al. 2017. pmdarima: arima estimators for Python. [Online; accessed 9.1.2021]. (2017). http://www.alkaline-ml.com/pmdarima.

Iris Recognition Based on SIFT and SURF Feature Detection

Alenka Trpin
Faculty of Information Studies
Ljubljanska cesta 31A
8000 Novo mesto, Slovenia
alenka.trpin@fis.unm.si

Bernard Ženko
Department of Knowledge Technologies
Jožef Stefan Institute
1000 Ljubljana, Slovenia
bernard.zenko@ijs.si

ABSTRACT

Human iris recognition is generally considered to be one of the most effective approaches for biometric identification. Identification is required in numerous areas such as security (e.g., airports and other buildings, airports), identity verification (e.g., banking, electoral registration), criminal justice system. This paper presents an approach for iris image classification that is based on two popular algorithms for image feature construction Scale Invariant Feature Transform (SIFT) and Speed Up Robust Features (SURF). Both algorithms were used in combination with the bag of visual words approach to create descriptive image features that can be used by supervised machine learning methods and a set of standard machine learning methods (k-Nearest Neighbor, random forest, support vector machines and neural networks) were evaluated on publicly available iris data set.

KEYWORDS

Iris recognition, image classification, SIFT features, SURF features

1 INTRODUCTION

Biometrics is the science of determining a person's identity and is an important approach for forensic and security identity management. Face, fingerprints, voice and iris are the most commonly used biometrics identifiers for personal identification. They provide characteristics in terms of personal appearance. The biometric system first scans the biometric characteristic, and then, typically based on a library of scans or classification model identifies the person [5].

Typical iris recognition system consists of four key modules: (1) image pre-processing, where the system detects the boundary of the pupil and the outer iris, (2) normalization, where the inner and outer circle parameters obtained from iris localization are given as input. Then, a transformation from polar to Cartesian coordinates is applied which maps the circle (iris) into a rectangle. (3) Feature extraction, where a feature vector is generated using different filters, and (4) comparison, based on different distances (Hamming distance in specific cases) between pairs of transformed iris images and the corresponding masks [10]. The comparison step nowadays frequently implemented with a machine learned classification model.

This work firs uses Scale Invariant Feature Transform (SUFT) and Speed Up Robust Features (SURF) algorithms to extract image keypoints or descriptors and then the bag of visual words to generate image features that can be used by standard supervised machine learning methods. We evaluate our method on a publicly available iris image dataset.

2 RELATED WORK

Iris recognition is frequently used for gender recognition and personal biometric authentication [6, 8, 9]. Ali et. al. applied contrast-limited adaptive histogram equalization to the normalized image. They used SURF and investigated the necessity of iris image enhancement based on the CASIA-Iris-Interval dataset [1]. Păvăloi and Ignat present experiments carried out with a new approach for iris image classification based on matching SIFT on iris occlusion images. They used the UPOL iris dataset to test their methods [6]. Bansal and Sharma use a statistical feature extraction technique based on the correlation between adjacent pixels, which was combined with a 2-D Wavelet Tree feature extraction technique to extract significant features from iris images. support vector machines (SVM) were used to classify iris images into male or female classes [2]. Salve et. al. used an artificial neural network and SVM as a classifier for iris patterns. Before applying the classifier, the region of interest, i.e., the iris region, is segmented using a Canny edge detector and a Hough transform. The eyelid and eyelash effect are kept to a minimum. A Daugman rubber-plate model is used to normalise the iris to improve computational efficiency and appropriate dimensionality. Furthermore, the discriminative feature sequence is obtained by feature extraction from the segmented iris image using 1D Log Gabor wavelet [14]. Adamović et. al. applied an approach that classifies biometric templates as numerical features in the CASIA iris image collection. These templates are generated by converting a normalised iris image into a one-dimensional fixed-length code set, which is then subjected to stylometric feature extraction. The extracted features are further used in combination with SVM and random forest (RF) classifiers [15].

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

3 METHODOLOGY

Our iris recognition approach combines image feature generation algorithms SIFT, SURF, bags of visual words model and standard supervised machine learning classification methods. In the following subsections we briefly describe each of these components, and then explain how these components are combined together.

3.1 SIFT

The SIFT algorithm detects a set of local features in an image. These features represent local areas of the image, and the algorithm also computes their description in a form of a vector. The algorithm proceeds in several stages. The first stage of computation is scale-space extrema detection which searches over all scales and image locations. It employs the so-called difference of Gaussian function to identify potential interest points that are invariant to scale and orientation. The second stage localizes each candidate at a location. Keypoints are extracted by detecting scale space extrema. The main idea behind the scale space extrema detection is to identify stable features which are invariant to changes in scale and viewpoint. At this point the keypoint descriptors are extracted [4, 6]. In essence, SIFT describes each image with a set of keypoints, and each keypoint is described with a vector of dimension 128. It is worth mentioning that SIFT can detect different numbers of keypoints in different images.

3.2 SURF

The SURF algorithm is based on similar ideas as SIFT, but their implementation is different. It can be used for similar tasks as SIFT, but it is faster, and produces highly accurate results when provided appropriate reference images. Instead of difference of Gaussian function, SURF uses approximate Laplacian of Gaussian images and a box filter. Determinants of the Hessian matrix are then used to detect the keypoints. A neighbourhood around the key point is selected and divided into sub-regions and then for each sub-region the wavelet responses are taken and represented to get SURF feature descriptor [1, 4]. In the end, each image is again represented with a set of keypoints, which are described with vectors.

3.3 Bag of Visual Words

The *bag of visual words* (BoVW) approach can be used for transforming or tokenizing keypoint-based image features, such as SIFT or SURF, into a fixed number of features, which is typically required by supervised machine learning methods. At first generates a visual word vocabulary from a (training) set of images, and then describes each image with these visual words. The visual word vector of an image contains the presence or absence information of each visual word in the image. In case of SIFT or SURF keypoints, for example, the visual word vector contains numbers of keypoints in an image that are similar to a given visual word. The process for extracting BoVW features from images involves the following steps: automatically detect regions or points of interest, compute local descriptors over those points (in our case, this means employing SIFT or SURF algorithm), quantize the descriptors into words to form the visual vocabulary, for example with a clustering algorithm, and find the occurrences in the image of each specific visual word in the vocabulary (generate a vector of visual word frequencies) [15]. It is worth mentioning that a specific BoVW model is based on a given training dataset and it only includes visual words that appear in the training images.

3.4 Classification Methods

The image classification phase of image analysis can be in principle performed with any machine learning method for classification. We have decided to evaluate a diverse set of standard methods, which we briefly describe in the following paragraphs.

The kNN is a supervised method that can be used for classification and regression. It is a simple algorithm where the classification of new instances is based on the majority class of the k closest training examples. The closeness is measured with a distance measure, which is usually Euclidean, Minkowski or Manhattan distance [9].

RF is a supervised learning algorithm based on the ensemble principle of using decision trees as the basic classifier and creating a learning model by combining multiple decision trees. The main idea of the RF classifier is to create multiple decision trees using a bootstrapped sampling method and introduce randomness in the individual tree building process. The class label of a new example is determined by majority voting of all trees in the ensemble [11].

The neural networks (NN) consist of several layers of simple units (neurons), which are simple functions with weight and bias parameters. Each neuron in one layer is connected to all neurons in the next layer by a process called back-propagation, and uses gradient descent to measure the rate of change of the loss function (e.g. Cross-Entropy loss). NN can have different structures, but typically have an input layer, one or more hidden layers and an output layer. Each of these layers contain one or more neurons [9, 12, 13]. In this work, we used the adam solver function because it is fast and gives good results. It is an optimisation algorithm that uses running averages of the two gradients and other moments of the gradients [13].

For the activation function, we use the logistic or sigmoid activation function. This determines how nodes in the network layer convert a weighted sum of input data into output data. The logistic or sigmoid activation function accepts any real value as input and the output values are from 0 to 1 [12].

Support vector machines (SVM) is a discriminant technique which means that the classification function takes a data point and assigns it to one of the different classes of the classification task. SVM transform the original data with a kernel function in a hyperspace, and then tries to find a hyperplane that distinguishes the two classes optimally. This hyperplane is defined with support vectors and distances between support vectors are maximised. SVM is very effective method for high dimensional problems [2, 14].

3.5 Our Method

Our approach for iris image classification is a based on the bag of visual words model, and we use either SIFT or SURF algorithm for image keypoint detection. In the training phase we perform the following steps.

1. For each image i, the SIFT or SURF algorithm is run, which detects K_i keypoints (each keypoint has $D = 128$ dimensions).
2. We collect keypoints from all training images, that is, $\sum_{i=1}^{n} K_i$ keypoints.
3. We cluster the above set of keypoints with the k-means clustering algorithm. Based on preliminary experimets we decided to use k = 500. The clusters, or their centroids, represent the visual words for our problem of iris recognition.
4. Now, we use the clustering model to assign each keypoint in an image to its nearest centroid (visual word) and sum up the occurrences of these visual words for each image. We end up with image descriptions, where each image is described with a vector of length k.
5. The dataset derived in the previous step can now be used to train a classification model with an arbitrary machine learning method. In our experiments, we have used four methods: k-Nearest Neighbor, Random Forest, Support Vector Machines and Neural Networks.

In the classification phase, when we need to classify a new image, we need to perform three steps.

1. Run the SIFT or SURF algorithm on the new image to detect keypoints (analogous to step 1 in training).
2. Use the clustering model to assign each keypoint to its nearest centroid and sum up their occurrences to derive visual words vector (analogous to step 4 in training).
3. Classify the image with the trained classification model.

We have performed experiments with two keypoint detection algorithms (SIFT and SURF) and four classification algorithms (kNN, RF, SVM and NN), and the results are presented in the next section.

4 RESULTS

For evaluating our approach, we have used the Ubiris.v1 dataset (http://iris.di.ubi.pt/ubiris1.html). It contains 1865 images of 200 x 150 resolution in 24-bit colours. They are grouped in two subsets: the first contains 1205 images in 241 classes and the second one contains 660 images in 132 classes. Images in the first subset have minimal noise factors, especially those related to reflections, luminosity, and contrast, because they were captured inside a dark room. The second subset of images was collected in a less controlled setting to introduce natural luminosity variation. This resulted in more heterogeneous images with included reflections, contrast, luminosity and focus problems. Images collected at this stage simulate the ones captured by a vision system without or with minimal active participation from the subjects [7].

These two subsets of images do not have the same classes. For our experiments we used the examples belonging to a subset of all classes: for the small subset we have selected 7 (the first seven classes) and for the big subset we have selected 127 classes (the first 127 classes). In the resulting datasets the examples were evenly distributed among the selected classes.

In our experiments we have used available Python implementations of included algorithms (scikit-learn for machine learning) with their default parameters, except the following:

- k-means: k=500,
- kNN: k=15, Euclidean metric,
- RF: number of estimators = 100,
- SVM: linear kernel function,
- NN: "adam" solver function, 8 hidden layers and 8 neurons, "logistic" activation function.

The classification accuracy was evaluated with 5-fold stratified cross validation. The results are presented separately for the small and big Ubiris.v1 datasets in Table 1 and 2, respectively.

Table 1: Classification accuracy on the small dataset with standard deviation

classifier/keypoint method	SIFT	SURF
kNN	0,37 ± 0,0	0,46 ± 0,0
RF	0,43 ± 0,06	0,63 ± 0,0
SVM	0,67 ± 0,0	0,86 ± 0,0
NN	0,63 ± 0,0	0,77 ± 0,0

The baseline accuracy for the small data set is 0.14 (i.e., 1/number of classes=1/7), and in Table 1 we can see that all instantiations of our method give better results than chance. The NN and SVM classifiers perform much better than RF and especially kNN. Comparing the keypoint detectors, we can see that SURF gives consistently better results than SIFT, although the difference is not very large. The results on the big dataset are, as expected, worse. The default accuracy in this case is 0.0079 (i.e., 1/127), and again all instantiations of our method give better results than chance. Again, SVM and NN perform best, but for some reason, NN performs very poorly in combination with SURF keypoints. RF in this case performs only slightly worse than SVM, while kNN is much worse. Also, on this data we can see that SURF keypoints give somewhat better results than SIFT, the only exception is NN, where SURF fails.

In summary we can conclude that for iris recognition the more complex learning algorithms (SVM, NN) outperform simpler ones (kNN and even RF), and that the SURF algorithm slightly outperforms SIFT. However, we can also conclude that iris recognition is a hard problem, which would probably benefit from application of state-of-the-art deep learning approaches.

Table 2: Classification accuracy on the big dataset with standard deviation

classifier/keypoint method	SIFT	SURF
kNN	0,02 ± 0,025	0,06 ± 0,039
RF	0,1 ± 0,018	0,11 ± 0,014
SVM	0,08 ± 0,039	0,13 ± 0,014
NN	0,17 ± 0,01	0,25 ± 0,005

To investigate whether any of the observed differences is statistically significant, we applied Friedman and Nemenyi tests as recommended in [8]. The results in the form of an average rank diagram with the estimated critical distance is presented in Figure 1 for big dataset and Figure 2 for small dataset.

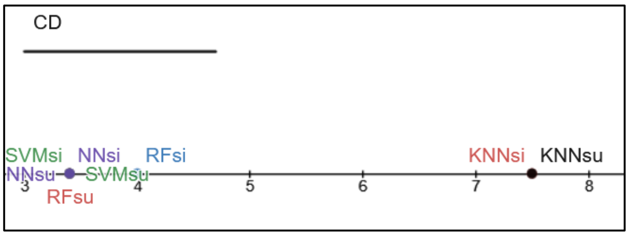


Figure 1: Average rank diagram with the estimated critical distance for the evaluated methods (small dataset)

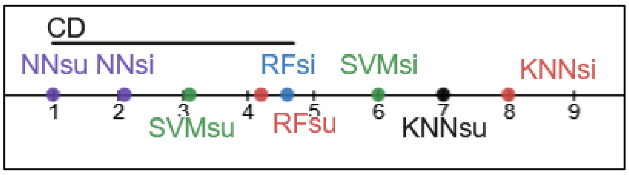


Figure 2: Average rank diagram with the estimated critical distance for the evaluated methods (big dataset)

The critical value for the eight classifiers and a confidence level of 0.05 is 3.031, the critical distance is CD = 4.695605.

Based on the size of CD we can only claim that the top of ranked methods and significantly better than the low ranked ones. For example, NN-SURF, NN-SIFT and SVM-SURF are better than KNN-SIFT. On the other hand, the differences among neighboring methods on the diagram are not significant.

5 CONCLUSION

The paper presents an evaluation of a typical bag of visual words approach on a specific dataset for human iris recognition. The results show that iris recognition is a relatively hard task and in order to improve the accuracy we would need a dataset with more examples of each class. In the future work we plan to evaluate additional feature extractors, like Oriented FAST and Rotated BRIEF (ORB) or Local Binary Pattern (LBP), and, given their success in image recognition in general, also convolutional neural networks approaches. With the latter, we will be especially interested in evaluating and comparing the performance vs. computational cost trade off.

REFERENCES

[1] Ali, H.S., Ismail, A.I., Farag, F.A. 2016. Speeded up robust features for efficient iris recognition. *SIViP* 10**,** 1385–1391 (2016).

[2] Atul Bansal, Ravinder Agarwal and R. K. Sharma, "SVM Based Gender Classification Using Iris Images," *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 425-429.

[3] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant keypoints. *International Journal of Computer Vision*, 60, 2, pp. 91-110.

[4] Ebrahim Karami, Siva Prasad, Mohamed Shehata. 2017. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *Newfoundland Electrical and Computer Engineering Conference.*

[5] Hájek J., Drahanský M. 2019. Recognition-Based on Eye Biometrics: Iris and Retina. In: Obaidat M., Traore I., Woungang I. (eds) Biometric-Based Physical and Cybersecurity Systems. Springer, Cham.

[6] Ioan Păvăloi and Anca Ignat. 2019. Iris Image Classification Using SIFT Features. 23rd International Conference on Knowledge-Based Systems and Intelligent Information & Engineering Systems, Elsevier. 159 (2019) 241–250.

[7] Hugo Pedro Proença and Luís A. Alexandre. 2005. UBIRIS: A noisy iris image database. *13th International Conference on Image Analysis and Processing - ICIAP 2005*, Springer, (Sept, 2005) 970-977.

[8] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res., 7, 1-30.

[9] Jiawei Han, Micheline Kamber and Jian Pei. 2012. *Data Mining: Concepts and Techniques*. (3rd ed.). The Morgan Kaufmann.

[10] John Daugman. 2004. How iris recognition works. IEEE Trans Cir-cuits Syst Video Technol 14(1): 21–30.

[11] Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1), 5-32.

[12] Saša Adamović, Vladislav Miškovic, Nemanja Maček, Milan Milosavljević, Marko Šarac, Muzafer Saračević, Milan Gnjatović. 2020. An efficient novel approach for iris recognition based on stylometric features and machine learning techniques, Future Generation Computer Systems, 107 (2020), 144-157.

[13] Shervin Minaee and Abdolrashidi Amirali. 2019. DeepIris: Iris Recognition Using a Deep Learning Approach.

[14] Sushilkumar S. Salve and S. P. Narote. 2016. Iris recognition using SVM and ANN. *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 474-478.

[15] Wadhah Ayadi, Wajdi Elhamzi, Imen Charfi, Mohamed Atri. 2019. A hybrid feature extraction approach for brain MRI classification based on Bag-of-words. *Biomedical Signal Processing and Control*, 48, 144-152.

Analyzing the Diversity of Constrained Multiobjective Optimization Test Suites

Aljoša Vodopija
aljosa.vodopija@ijs.si
Jožef Stefan Institute and
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Tea Tušar
tea.tusar@ijs.si
Jožef Stefan Institute and
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Bogdan Filipič
bogdan.filipic@ijs.si
Jožef Stefan Institute and
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

A well-designed test suite for benchmarking novel optimizers for constrained multiobjective optimization problems (CMOPs) should be diverse enough to detect both the optimizers' strengths and shortcomings. However, until recently there was a lack of methods for characterizing CMOPs, and measuring the diversity of a suite of problems was virtually impossible. This study utilizes the landscape features proposed in our previous work to characterize frequently used test suites for benchmarking optimizers in solving CMOPs. In addition, we apply the t-distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction approach to reveal the diversity of these test suites. The experimental results indicate which ones express sufficient diversity.

KEYWORDS

constrained multiobjective optimization, benchmarking, landscape feature, t-SNE

1 INTRODUCTION

Real-world optimization problems frequently involve multiple objectives and constraints. These problems are called *constrained multiobjective optimization problems* (CMOPs) and have been gaining a lot of attention in the last years [13]. As with other theoretically-oriented optimization studies, a crucial step in testing novel algorithms in constrained multiobjective optimization is the preparation of a benchmark test.

One of the key elements of a benchmark test is the selection of suitable test CMOPs [1]. A well-designed benchmark suite should include "a wide variety of problems with different characteristics" [1]. This way the benchmark problems are *diverse* enough to "highlight the strengths as well as weaknesses of different algorithms" [1]. However, until recently there existed only few and limited techniques proposed to explore CMOPs [13]. For this reason, the test suites of CMOPs were insufficiently understood and measuring their diversity was virtually impossible.

To overcome this situation, in our previous work [13], we experimented with various exploratory landscape analysis (ELA) techniques and proposed 29 landscape features to characterize CMOPs, including their *violation landscapes*—a similar concept as the fitness landscape where fitness is replaced by the *overall constraint violation*.

In this study, we employ the landscape features proposed in [13] to express and discuss the diversity of frequently used test suites of CMOPs. This is achieved by firstly computing the landscape features and then employing the t-distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique, to embed the 29-D CMOP feature space into the 2-D space. Note that due to space limitations, only selected results are shown in this paper. The complete results can be found online[1].

The rest of this paper is organized as follows. Section 2 provides the theoretical background. In Section 3, we present the landscape features and the t-SNE algorithm. Section 4 is dedicated to the experimental setup, while the results are discussed in Section 5. Finally, Section 6 summarizes the study and provides an idea for future work.

2 THEORETICAL BACKGROUND

A CMOP can be formulated as:

$$\begin{aligned} &\text{minimize} && f_m(x), && m = 1, \dots, M \\ &\text{subject to} && g_i(x) \le 0, && i = 1, \dots, I \end{aligned} \tag{1}$$

where $x = (x_1, \dots, x_D)$ is a *search vector*, $f_m : S \to \mathbb{R}$ are *objective functions*, $g_i : S \to \mathbb{R}$ *constraint functions*, $S \subseteq \mathbb{R}^D$ is a *search space* of dimension D, and M and I are the numbers of objectives and constraints, respectively.

If a solution x satisfies all the constraints, $g_i(x) \le 0$ for $i = 1, \dots, I$, then it is a *feasible* solution. For each of the constraints g_i we can define the *constraint violation* as $v_i(x) = \max(0, g_i(x))$. In addition, an *overall constraint violation* is defined as

$$v(x) = \sum_i^I v_i(x). \tag{2}$$

A solution x is feasible iff $v(x) = 0$.

A feasible solution $x \in S$ is said to *dominate* a solution $y \in S$ if $f_m(x) \le f_m(y)$ for all $1 \le m \le M$, and $f_m(x) < f_m(y)$ for at least one $1 \le m \le M$. In addition, $x^* \in S$ is a *Pareto-optimal solution* if there exists no $x \in S$ that dominates x^*. All feasible solutions represent a *feasible region*, $F = \{x \in S \mid v(x) = 0\}$. Besides, all nondominated feasible solutions form a *Pareto-optimal set*, S_o. The image of the Pareto-optimal set is the *Pareto front*, $P_o = \{f(x) \mid x \in S_o\}$. A connected component (a maximal connected subset with respect to the inclusion order) of the feasible region is called a *feasible component*, $\mathcal{F} \subseteq F$.

In [13], we introduced analogous terms from the perspective of the overall constraint violation. A *local minimum-violation solution* is thus a solution x^* for which exists a $\delta > 0$ such that $v(x^*) \le v(x)$ for all $x \in \{x \mid d(x^*, x) \le \delta\}$. If there is no other solution $x \in S$ for which $v(x^*) > v(x)$, then x^* is a

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

[1] https://vodopijaaljosa.github.io/cmop-web/

(global) minimum-violation solution. We denoted the set of all local minimum-violation solutions by F_l and called a connected component $\mathcal{M} \subseteq F_l$ a *local minimum-violation component.*

In order to express the modality of a violation landscape, we defined a local search procedure to be a mapping from the search space to the set of local minimum-violation solutions, $\mu : S \longrightarrow F_l$, such that $\mu(x) = x$ for all $x \in F_l$. A *basin of attraction* of a local minimum-violation component $\mathcal{M}$ and local search μ is then a subset of S in which μ converges towards a solution from $\mathcal{M}$, i.e., $\mathcal{B}(\mathcal{M}) = \{x \in S \mid \mu(x) \in \mathcal{M}\}$. The violation landscape is *unimodal* if there is only one basin in S and *multimodal* otherwise.

3 METHODOLOGY

3.1 ELA Features

The landscape features used in this study were introduced in our previous work [13] and can be categorized into four groups: space-filling design, information content, random walk and adaptive walk features. They are summarized in Table 1.

The space-filling design features are used to quantify the feasible components, the relationship between the objectives and constraints, and measure the feasibility ratio and proportion of boundary Pareto-optimal solutions. Next, the information content features are mainly used to express the smoothness and ruggedness of violation landscapes. They are derived by analyzing the entropy of sequences of overall violation values as obtained from a random sampling of the search space. Then, the random walk features considered in this study are used to quantify the number of boundary crossings from feasible to infeasible regions. They are used to categorize the degree of segmentation of the feasible region. Finally, features from the last group are derived from adaptive walks through the search space. They are used to describe various aspects of basins of attraction in the violation landscapes.

3.2 Dimensionality Reduction with t-SNE

The t-SNE algorithm is a popular nonlinear dimensionality reduction technique designed to represent high-dimensional data in a low-dimensional space, typically the 2-D plane [12]. First, it converts similarities between data points to distributions. Then, it tries to find a low-dimensional embedding of the points that minimizes the divergence between the two distributions that measure neighbor similarity—one in the original space and the other in the projected space. This means that t-SNE tries to preserve the local relationships between neighboring points, while the global structure is generally lost.

Finding the best embedding is an optimization problem with a non-convex fitness function. To solve it, t-SNE uses a gradient descent method with a random starting point, which means that different runs can yield different results. The output of t-SNE depends also on other parameters, such as the *perplexity* (similar to the number of nearest neighbors in other graph-based dimensionality reduction techniques), *early exaggeration* (separation of clusters in the embedded space) and *learning rate* (also called ε). The gradients can be computed exactly or estimated using the Barnes-Hut approximation, which substantially accelerates the method without degrading its performance [11].

4 EXPERIMENTAL SETUP

We studied eight suites of CMOPs which are most frequently used in the literature. These are CTP [2], CF [14], C-DTLZ [5], NCTP [7], DC-DTLZ [8], LIR-CMOP [3], DAS-CMOP [4], and MW [9]. In addition, we included also a novel suite named RCM [6]. In contrast to other suites which consist of artificial test problems, RCM contains 50 instances of real-world CMOPs based on physical models. Note that we actually used only 11 RCM problems, since only continuous and low-dimensional problems were suitable for our analysis. We considered three dimensions of the search space: 2, 3, 5. It is to be noted that large-scale CMOPs were not taken into account since the methodology described in Section 3 is not sufficiently scalable. This limits our results to low-dimensional CMOPs. Table 2 shows the basic characteristics of the studied test suites.

Table 1: The ELA features used to characterize CMOPs categorized into four groups: space-filling design, information content, random walk, and adaptive walk [13].

Space-filling design features	
$N_{\mathcal{F}}$	Number of feasible components
$\mathcal{F}_{\text{min}}$	Smallest feasible component
$\mathcal{F}_{\text{med}}$	Median feasible component
$\mathcal{F}_{\text{max}}$	Largest feasible component
$O(\mathcal{F}_{\text{max}})$	Proportion of Pareto-optimal solutions in $\mathcal{F}_{\text{max}}$
$\mathcal{F}_{\text{opt}}$	Size of the "optimal" feasible component
ρ_{F}	Feasibility ratio
ρ_{min}	Minimum correlation
ρ_{max}	Maximum correlation
$\rho_{\partial S_o}$	Proportion of boundary Pareto-optimal solutions
Information content features	
H_{max}	Maximum information content
ε_s	Settling sensitivity
M_0	Initial partial information
Random walk features	
$(\rho_{\partial F})_{\text{min}}$	Minimal ratio of feasible boundary crossings
$(\rho_{\partial F})_{\text{med}}$	Median ratio of feasible boundary crossings
$(\rho_{\partial F})_{\text{max}}$	Maximal ratio of feasible boundary crossings
Adaptive walk features	
$N_{\mathcal{B}}$	Number of basins
$\mathcal{B}_{\text{min}}$	Smallest basin
$\mathcal{B}_{\text{med}}$	Median basin
$\mathcal{B}_{\text{max}}$	Largest basin
$(\mathcal{B}_{\text{F}})_{\text{min}}$	Smallest feasible basin
$(\mathcal{B}_{\text{F}})_{\text{med}}$	Median feasible basin
$(\mathcal{B}_{\text{F}})_{\text{max}}$	Largest feasible basin
$\cup\mathcal{B}_{\text{F}}$	Proportion of feasible basins
$v(\mathcal{B})_{\text{med}}$	Median constraint violation over all basins
$v(\mathcal{B})_{\text{max}}$	Maximum constraint violation of all basins
$v(\mathcal{B}_{\text{max}})$	Constraint violation of $\mathcal{B}_{\text{max}}$
$O(\mathcal{B}_{\text{max}})$	Proportion of Pareto-optimal solutions in $\mathcal{B}_{\text{max}}$
$\mathcal{B}_{\text{opt}}$	Size of the "optimal" basin

For dimensionality reduction, we used the t-SNE implementation from the *scikit-learn* Python package [10] with default parameter values. That is, we used the Euclidean distance metric, random initialization of the embedding, perplexity of 30, early exaggeration of 12, learning rate of 200, the maximum number of iterations of 1000, and the maximum number of iterations without progress before aborting of 300. The gradient was computed by the Barnes-Hut approximation with the angular size of 0.5.

5 RESULTS AND DISCUSSION

The results obtained by t-SNE are shown in Figures 1 and 2. Specifically, the figures show the 2-D embedding of the 29-D

Table 2: Characteristics of test suites: number of problems, dimension of the search space D, number of objectives M, and number of constraints I. The characteristics of selected RCM problems are shown in parentheses.

Test suite	#problems	D	M	I
CTP [2]	8	*	2	2, 3
CF [14]	10	*	2, 3	1, 2
C-DTLZ [5]	6	*	*	1, *
NCTP [7]	18	*	2	1, 2
DC-DTLZ [8]	6	*	*	1, *
DAS-CMOP [4]	9	*	2, 3	7, 11
LIR-CMOP [3]	14	*	2, 3	2, 3
MW [9]	14	*	2, *	1–4
RCM [6]	50 (11)	2–34 (2–5)	2–5	1–29 (1–8)

*Scalable parameter.

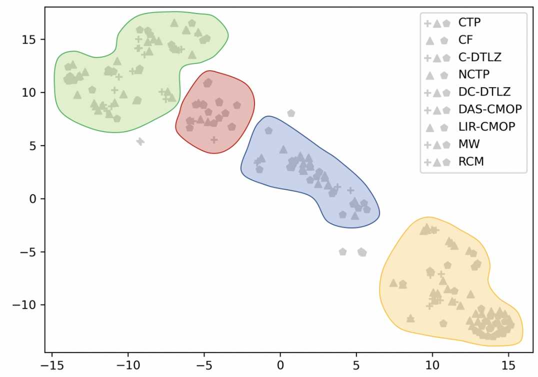


Figure 1: Embedding of the feature space as obtained by t-SNE. The four regions are depicted in green, red, blue, and orange. The points that are not contained in any region are considered to be outliers.

feature space consisting of the landscape features presented in Table 1. Each subfigure in Figure 2 corresponds to one of the test suites. For example, Figure 2a exposes the embedding of the CTP suite in blue, while the gray points correspond to the rest of the test suites. Points with a shape of a plus (+) correspond to CMOPs with two variables, points with a shape of a triangle (▲) to CMOPs with three variables, and points with a shape of a pentagon (⬟) to CMOPs with five variables.

An additional analysis shows that the embedding of the feature space can be, based on the corresponding characteristics, split into four regions: green, red, blue and yellow (Figure 1). The green region corresponds to CMOPs with severe violation multimodality, small basins of attraction, and rugged violation landscapes. The red region corresponds to CMOPs with moderate violation multimodality, rugged violation landscapes, and small feasibility ratios. The blue region corresponds to relatively low violation multimodality, rugged violation landscapes, small feasibility ratios, and positive correlations between objectives and constraints. Finally, the yellow region corresponds to unimodal CMOPs with large feasible components, smooth violation landscapes, and large feasible regions.

As we can see from Figure 2a, almost all CTP problems are located in the orange region. Therefore, many relevant characteristics are poorly represented by CTP, e.g., violation multimodality, small feasibility ratios, etc. Similarly, NCTP fails to sufficiently represent severe multimodality since it contains no problems from the green region (Figure 2d). On the other hand, DC-DTLZ, LIR-CMOP, and MW are biased towards highly multimodal violation landscapes or those with small basins of attraction (Figure 2e, Figure 2g, and Figure 2h). Nevertheless, MW is one of the most diverse suites considering other characteristics (Figure 2h).

The C-DTLZ and DAS-CMOP suites are mainly located in the green and orange regions and fail to sufficiently represent the characteristics of the red and blue regions.

Finally, the results show that CF and RCM are well spread through the whole embedded feature space (Figure 2b and Figure 2i). As we can see, they have at least one representative CMOP instance in each region. Therefore, CF and RCM are the most diverse test suites according to the employed landscape features.

6 CONCLUSIONS

In this paper, we analyzed the diversity of the frequently used test suites for benchmarking optimizers in solving CMOPs. For this purpose, we considered 29 landscape features for CMOPs that were proposed in our previous work. In addition, the t-SNE algorithm was used to reduce the dimensionality of the feature space and reveal the diversity of the considered test suites.

The experimental results show that the most diverse test suites of CMOPs according to the applied landscape features are CF and RCM. Indeed, they include the widest variety of CMOPs with different characteristics. In addition, MW also proved to be a diverse suite except for unimodal CMOPs. Nevertheless, we suggest to consider CMOPs from various test suites for benchmarking optimizers in constrained multiobjective optimization.

One of the main limitations of our study is that only low-dimensional CMOPs were used in the analysis. Therefore, we were unable to adequately address the issue of scalability. For this reason, a crucial task that needs to be addressed in the feature is the extension of this work to large-scale CMOPs.

ACKNOWLEDGMENTS

We acknowledge financial support from the Slovenian Research Agency (young researcher program and research core funding no. P2-0209). This work is also part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement no. 692286.

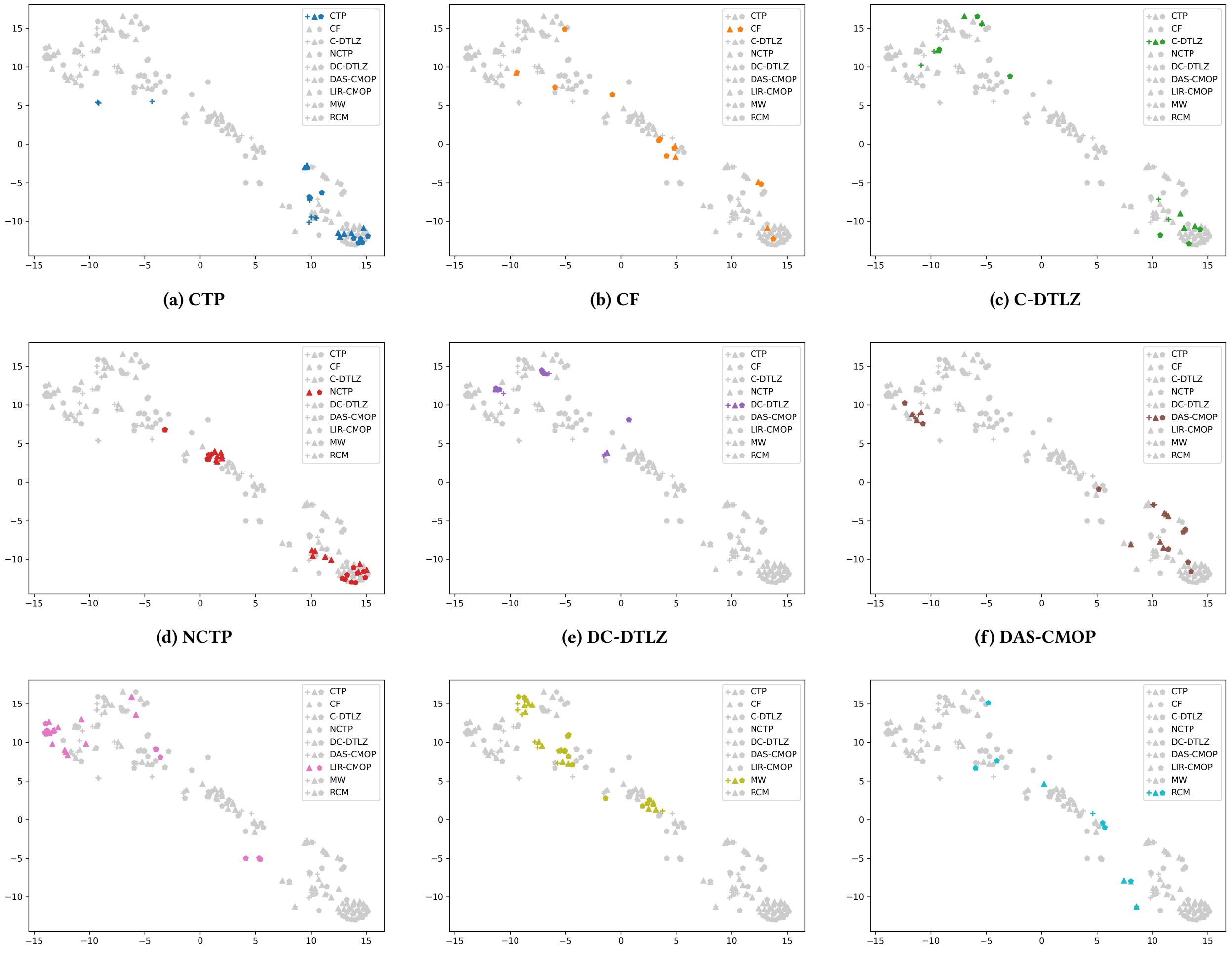
REFERENCES

[1] T. Bartz-Beielstein, C. Doerr, J. Bossek, S. Chandrasekaran, T. Eftimov, A. Fischbach, P. Kerschke, M. López-Ibáñez, K. M. Malan, J. H. Moore, B. Naujoks, P. Orzechowski, V. Volz, M. Wagner, and T. Weise. Benchmarking in optimization: Best practice and open issues. arXiv:2007.03488v2, (2020).

[2] K. Deb, A. Pratap, and T. Meyarivan. 2001. Constrained test problems for multi-objective evolutionary optimization. In *Evolutionary Multi-Criterion Optimization (EMO 2001)*, 284–298.

[3] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman. 2019. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput.*, 23, 23, 12491–12510. DOI: 10.1007/s00500-019-03794-x.

[4] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman. 2019. Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evol. Comput.*, 28, 3, 339–378. DOI: 10.1162/evco-a-00259.



(a) CTP

(b) CF

(c) C-DTLZ

(d) NCTP

(e) DC-DTLZ

(f) DAS-CMOP

(g) LIR-CMOP

(h) MW

(i) RCM

Figure 2: Embedding of the feature space as obtained by t-SNE. Each subfigure exposes the embedding of a selected suite.

[5] H. Jain and K. Deb. 2014. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.*, 18, 4, 602–622. DOI: 10.1109/TEVC.2013.2281534.

[6] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das. 2020. A Benchmark-Suite of Real-World Constrained Multi-Objective Optimization Problems and some Baseline Results. Technical report. Indian Institute of Technology, Banaras Hindu University Campus, India.

[7] J. P. Li, Y. Wang, S. Yang, and Z. Cai. 2016. A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization. In *IEEE Congress on Evolutionary Computation (CEC 2016)*, 4175–4182. DOI: 10.1109/CEC.2016.7744320.

[8] K. Li, R. Chen, G. Fu, and X. Yao. 2019. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. Evol. Comput.*, 23, 2, 303–315. DOI: 10.1109/TEVC.2018.2855411.

[9] Z. Ma and Y. Wang. 2019. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans. Evol. Comput.*, 23, 6, 972–986. DOI: 10.1109/TEVC.2019.2896967.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, 12, 2825–2830.

[11] L. van der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.*, 15, 1, 3221–3245.

[12] L. van der Maaten and G. Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9, 2579–2605.

[13] A. Vodopija, T. Tušar, and B. Filipič. Characterization of constrained continuous multiobjective optimization problems: A feature space perspective. arXiv:2109.04564, (2021).

[14] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. 2008. Multiobjective optimization test instances for the CEC 2009 special session and competition. Technical report CES-487. The School of Computer Science and Electronic Engieering, University of Essex, UK.

Corpus KAS 2.0: Cleaner and with New Datasets

Aleš Žagar, Matic Kavaš, Marko Robnik-Šikonja
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia
{ales.zagar,matic.kavas,marko.robnik}@fri.uni-lj.si

ABSTRACT

Corpus of Academic Slovene (KAS) contains Slovene BSc/BA, MSc/MA, and PhD theses from 2000 - 2018. We present a cleaner version of the corpus with added text segmentation and updated POS-tagging. The updated corpus of abstracts contains fewer artefacts. Using machine learning classifiers, we filled in missing research field information in the metadata. We used the full texts and corresponding abstracts to create several new datasets: monolingual and cross-lingual datasets for long text summarization of academic texts and a dataset of aligned sentences from abstracts in English and Slovene, suitable for machine translation. We release the corpora, datasets, and developed source code under a permissible licence.

KEYWORDS

KAS corpus, academic writing, machine translation, text summarization, CERIF classification

1 INTRODUCTION

The Corpus of Academic Slovene (KAS 1.0)[1] is a corpus of Slovenian academic writing gathered from the digital libraries of Slovenian higher education institutions via the Slovenian Open Science portal[2] [3]. It consists of diploma, master, and doctoral theses from Slovenian institutions of higher learning (mostly from the University of Ljubljana and the University of Maribor). It contains 82,308 texts with almost 1.7 billion tokens.

The KAS texts were extracted from the PDF formatted files, which are not well-suited for the acquisition of high-quality raw texts. For that reason, the KAS corpus is noisy. Our analysis showed that most original texts contain tables, images, and other kinds of figures which are transformed into gibberish when converted from the PDF format. The extracted figure captions also do not give any helpful information. Some texts contain front or back matter (for example, a table of contents at the beginning or references at the end), which shall not be present in the main text body.

The Corpus of KAS abstracts (KAS-Abs 1.0)[3] contains 47,273 only Slovene, 49,261 only English, and 11,720 abstracts in both languages. We observed several shortcomings of this corpus. A vast majority of abstracts contain keywords or the word "Abstract" somewhere in the abstract text. Many texts contain other kinds of meta-information, e.g., the name of the author or supervisor and the title of the thesis. Several corpus entries contain English and Slovene abstracts in the same unit, only one of them wrongly marked to contain both abstracts or switched Slovene and English abstracts. Several entries did not contain the abstract; instead, there was front or back matter like copyright statement, table of contents, list of abbreviations etc.

Our analysis has shown that the corpora can be improved in many aspects. Besides addressing the above-mentioned weaknesses, the main improvements in the updated KAS 2.0 and KAS-Abs 2.0 corpora are chapter segmentation and improved metadata with machine learning methods (described in Sections 2 and 3). A further motivation for our work is the opportunity to extract valuable new datasets for text summarization (monolingual and cross-lingual) and a sentence-aligned machine translation dataset created from matching Slovene and English abstracts (see Section 4). We present conclusions and ideas for further improvements in Section 5.

2 UPDATES: KAS 2.0 AND KAS-ABS 2.0

We first describe methods for extracting text and abstracts from PDF, followed by the differences between the versions 1.0 and 2.0 of corpora.

2.1 Extraction of Text Body

As many texts in corpora version 1.0 contained several hard to fix faults (like gibberish due to extracted tables and figures), we decided to extract texts once again from the PDFs. We used the pdftotext tool, which is a part of the poppler-utils. The software proved to be accurate and reliable. Its important feature is keeping the original text layout and excluding the areas where we detected figures, tables, and other graphical elements.

In the first step, we converted PDF files to images, one page at a time and used the OpenCV computer vision library to detect text and non-text areas. We marked the text areas on each page. For each document, we also calibrated the size of the header and footer areas and removed them from the text areas together with the page numbers. In this process, we removed 2,467 out of the original 91,019 documents due to the documents containing less than 15 pages or some unchecked exceptions in the code.

Next, we searched for the beginning and the end of the main text body. We observed that practically all bodies start with some variation of the Slovene word "Uvod" (i.e. introduction). If we found the beginning, we searched for the ending in the same way but with different keywords (viri, literatura, povzetek, etc). For texts with found beginning and end, the areas were clipped and the extracted texts were normalized. The normalization included handling Slovene characters with the caret (č, š, ž), ligattures (tt, ff, etc.), removal of remaining figure and table captions, and empty lines. The obtained text was segmented into the structure extracted from the table of contents. We matched headings in the text with the entries in the table of contents and used page numbers as guidelines. We ended with 83,884 successfully extracted documents.

[1] https://www.clarin.si/repository/xmlui/handle/11356/1244
[2] https://www.openscience.si/
[3] https://www.clarin.si/repository/xmlui/handle/11356/1420

Information Society 2021, 4–8 October 2021, Ljubljana, Slovenia

2.2 Extraction of Abstracts

We tried to improve the KAS-abstracts corpus by cleaning the existing documents and extracting the abstracts directly from the PDFs. An initial analysis of existing texts showed different formattings (71 different organizations publish the works in the KAS corpus). We identified five major patterns of problems and created scripts for resolving them. This produced approximately 40,000 cleaned texts while 20,000 were still problematic. The direct extraction from the PDFs followed the same procedure as for the main text body (described above). We considered figures, headers, footers, page numbers, keywords, meta-information, abstract placement at the beginning and end of the documents, multiple abstracts of different lengths, etc. This resulted in 71,567 collected Slovene abstracts. A similar procedure was applied to English abstracts and yielded 53,635 abstracts.

2.3 Differences from Version 1.0 to 2.0

Besides cleaner texts, excluded gibberish from figures and tables, and excluded front- and back-matter, the most important difference between KAS versions 1.0 and 2.0 is that the texts are segmented by structure, i.e. by headings. Unfortunately, some documents present in the original KAS were lost due to the different extraction, and for some documents appearing only in version 2.0, there is no metadata.

KAS-abstracts is greatly improved and no longer contains large quantities of unusable text and different artefacts (e.g., metadata, keywords, or front- and back-matter). Again, for some abstracts present only in version 2.0, there is no metadata. Still, they are usable for several tasks, including machine translation studies. Table 1 gives the quantitative overview of the obtained body texts and abstracts.

Table 1: Statistics of the obtained body texts and abstracts in version 2.0 of the KAS corpora.

	Sum	Same as in 1.0	Missing from 1.0	With metadata
Slo abstracts	71,567	56,610	2,383	67,533
Eng abstract	53,635	44,685	16,296	50,674
Body text	83,884	79,320	2,988	79,320

3 SUB-CERIF CLASSIFICATION

CERIF (Common European Research Information Format) is the standard that the EU recommends to member states for recording information about the research activity[4]. The top level has only five categories (humanities, social sciences, physical sciences, biomedical sciences, and technological sciences). In comparison, the lower level distinguishes 363 categories. As Slovene libraries use the UDC classification, in the KAS corpus 1.0, only 17% of the documents also contain the CERIF and sub-CERIF codes in their metadata. These are mapped from UDC codes by the heuristics produced by the Slovene Open Science Portal. Below, we describe how we automatically annotated documents with missing sub-CERIF codes using a machine learning approach.

We build a dataset for automatic annotation of sub-CERIF codes from the body texts of the documents. A document may have more than one sub-CERIF code, which means that classes are not mutually exclusive. Thus, we tackle a multi-label classification problem. In the corpus, there are 13,738 documents with high confidence levels of CERIF codes which we use in machine learning. Our dataset contains 64 labels out of 363 possible. We used 10% or 1374 samples as the test set and the remaining 90% as the training set.

As several studies have shown that recent neural embedding approaches are not yet competitive with standard text representations in document level tasks, we decided to use standard Bag-of-Words representation with TF-IDF weighting. In the preprocessing step, we lemmatized texts using CLASSLA lemmatizer[5] and removed stop-words[6] and punctuation.

We compared four classifiers. For logistic regression (LR), k-nearest neighbours (KNN), and support vector machines (SVM), we used Scikit-learn [6], and for the multi-layer perceptron (MLP), we tried Keras implementation. For the first three, we preliminary tried several different parameter values but found that they perform the best with the default ones. The MLP neural network consists of one hidden layer with 256 units, sigmoid activation function on hidden and output layers, Adam optimizer [5] with an initial learning rate of 0.01, and binary cross-entropy as a loss function. We used the early stopping (5 consecutive epochs with no improvement) and reduced the learning rate on the plateau (halving learning rate for every 2 epochs with no improvement) as callbacks during the learning process.

In Table 2, we report pattern accuracy and binary accuracy of the trained classifiers. A model predicts a correct pattern if it assigned all true sub-CERIF codes to a document. For binary accuracy, a model predicts a sub-CERIF code correctly if it assigns a true single sub-CERIF code to the document. For example, let us assume that we have four sub-CERIF codes and an example with a label sequence '1010'. If a model predicts '1010', it receives 100% for both pattern and binary accuracy. If a model predicts '0010', it gets 0% pattern accuracy and 75% binary accuracy since it misclassified only the first label.

Table 2: Results on the sub-CERIF multi-label classification task. The best result for each metric is in bold.

Algorithm	Binary accuracy	Pattern accuracy
LR	98.48	38.36
KNN	98.52	43.75
SVM	**98.68**	**47.82**
MLP	98.66	46.58

Using the pattern accuracy metric, SVM and MLP are significantly better than KNN and LR. LR is the worst performing model, and KNN is in the middle. SVM is the best, and MLP is behind for 1.24 points. We assume that we do not have enough data for MLP to beat SVM. It is difficult to assess the models regarding binary accuracy. In the test set, we have 761 examples with 1 label, 466 with 2 labels, 107 with 3 labels, 26 with 4 labels, 10 with 5, and 4 with 6. A dummy model that predicts all zeros achieves binary accuracy of 97.51. All our models are better than this baseline, and their ranks correspond with the pattern accuracy.

We conclude that given 64 labels and 10k training instances, our best model (SVM) correctly predicts almost half of them, which is a useful result.

[4] https://www.dcc.ac.uk/resources/metadata-standards/cerif-common-european-research-information-format

[5] https://github.com/clarinsi/classla

[6] We used the list from https://github.com/stopwords-iso/stopwords-sl

4 NEW DATASETS

We created two types of new datasets, described below: summarization datasets and machine translation datasets.

4.1 Summarization Datasets

We created two new datasets appropriate for *long-text summarization* in the monolingual and cross-lingual settings. The monolingual slo2slo dataset contains 69,730 Slovene abstracts and Slovene body texts and is suitable for training Slovene summarization models for long texts. The cross-lingual slo2eng dataset contains 52,351 Slovene body texts and English abstracts. It is suitable for the cross-lingual summarization task.

4.2 Machine Translation Datasets

For the creation of a sentence-aligned *machine translation dataset*, we used the neural approach proposed by Artetxe & Schwenk [1]. The main difference to other text alignment approaches is in using margin-based scoring of candidates in contrast to a hard threshold with cosine similarity. We improved the approach by replacing the underlying neural model. Instead of BiLSTM-based LASER [2] representation, we used the transformer-based LaBSE [4] sentence representation, which has significantly improved average bitext retrieval accuracy. We used the implementation from UKPLab[7]. This approach requires a threshold that omits candidate pairs below a certain value. This value represents a trade-off between the quantity and quality of aligned pairs. The higher the threshold, the better the quality of alignments, but more samples are discarded.

In text alignment, sentences do not always exhibit one-to-one mapping: a source sentence can be split into two or more target sentences and vice versa. To address the problem, we iteratively ran the alignment process until all sentences above the chosen threshold were assigned to each other. In cases of more than one sentence assigned to a single sentence, we merged them and thus created a translation pair.

We manually inspected the alignments consisting of more than one sentence in either source or target text on a small subset of abstracts. We observed that a merging process produces better results than imposing a restriction allowing only the one-to-one mapping. In Table 4, we present an example of the alignment. The first column represents a margin-based score. If an aligned pair contains more than one sentence in the source or target, the score consists of the average margin-based score between a single sentence and multiple sentences. The last column is an indicator of whether merging was applied.

We used the ratio variant of margin-based scoring and set the default threshold to 1.1. We manually tested the alignment on our internal dataset. From 2015 examples, we successfully aligned 2002 of them (99.3%), misaligned 1 (0.1%), and omitted 12 of them (0.6%). The analysis of 12 omitted cases showed that some pairs do not match each other or are not accurate translations of each other, e.g., a large part of the original sentence is omitted, phrases are only distantly related, etc. However, approximately half of the 12 cases shall be aligned, which means that our model works very well, but conservatively and may fail for free translation pairs.

With the default value of the threshold (1.1), we produced 496.102 sentence pairs. We believe the threshold is strict enough to produce good-quality dataset (especially if compared to many other sentence alignments in existing translation datatsets). However, if one would prefer even more certain alignment, the value of the threshold can be further increased at the expense of less sentences in the datatset. We released three such datasets that reflect a trade-off between quality and quantity of the data. The sizes of the obtained datasets are available in Table 3.

Table 3: Size of the machine translation datasets based on the margin-based quality threshold.

Dataset	Threshold	Size
Normal alignment	1.1	496,102
Strict alignment	1.2	474,852
Very strict alignment	1.3	425,534

5 CONCLUSIONS

In this work, we created version 2.0 of Corpus KAS and Corpus KAS-Abstracts. We cleaned the texts and abstracts, introduced the text segmentation based on its structure, and improved the metadata. We created two new long text summarization datasets and a dataset of aligned sentences for machine translations. The latest versions of corpora and datasets are available on the CLARIN.SI. The corpora are annotated with the CLASSLA tool and released in txt, JSON and TEI formats. The source code for producing the new versions of the corpora[8] and the created datasets are publicly available[9] .

In future work, the extraction of metadata for entries where they are missing would be beneficial. There could be further improvements in cleaning the texts, and this would increase the number of available documents. When the corpora are extended with data post-2018, the software might need further modifications due to new formats and templates used in the academic works. Further experiments on the created MT datasets would clarify the setting of parameters and show if current MT systems benefit more from better quality or larger quantity of data.

ACKNOWLEDGMENTS

The research was supported by CLARIN.SI (2021 call), Slovenian Research Agency (research core funding program P6-0411), Ministry of Culture of Republic of Slovenia through project Development of Slovene in Digital Environment (RSDO), and European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). We thank Tomaž Erjavec (JSI, Department of Knowledge Technologies) for providing data access and his assistance in building TEI format of the corpus.

REFERENCES

[1] Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3197–3203.

[7] https://github.com/UKPLab/sentence-transformers/blob/master/examples/applications/parallel-sentence-mining/bitext_mining.py

[8] https://github.com/korpus-kas

[9] KAS 2.0: https://www.clarin.si/repository/xmlui/handle/11356/1448
KAS-Abs 2.0: https://www.clarin.si/repository/xmlui/handle/11356/1449
Summarization datasets: https://www.clarin.si/repository/xmlui/handle/11356/1446
MT datasets: https://www.clarin.si/repository/xmlui/handle/11356/1447

Table 4: Examples from sentence-aligned Slovene-English abstracts.

Score	Slovene source sentence	English target sentence	Mrg
1.670	Moški pa pogosteje opravljajo opravila, ki se tičejo mehanizacije na kmetiji.	Men, however, often perform tasks related to machinery on the farm.	No
1.612	Zanimala nas je tudi prisotnost tradicionalnih vzorcev pri delu.	Additionally, I have also focused on the presence of traditional work patterns.	No
1.520	Želeli smo izvedeti, ali se kmečke ženske počutijo preobremenjene, cenjene in kako preživljajo prosti čas (če ga imajo).	I wanted to know whether rural women feel overwhelmed or valued, and how they spend their free time (if they have it).	No
1.441	Dotaknili smo se tudi problemov, s katerimi se srečujejo kmečke ženske med javnim in zasebnim življenjem.	Moreover, I have tackled the problems that rural women face when it comes to their public and private life.	No
1.437	Na koncu teoretičnega dela smo opisali še predloge za izboljšanje položaja kmečkih žensk v družbi.	At the end of the theoretical part, I have denoted further proposals for improving the situation of rural women in today's society.	No
1.388	V diplomskem delu obravnavamo položaj žensk v kmečkih gospodinjstvih v Sloveniji.	The thesis deals with the situation of women in rural households of Slovenia.	No
1.354	V empiričnem delu pa smo s pomočjo anketnega vprašalnika, na katerega so kot respondentke odgovarjale kmečke ženske, ugotavljali, kako je delo na kmetiji porazdeljeno med spoloma.	In the empirical part, I have conducted a survey on peasant women to determine the gender division of farm labour.	No
1.271	V teoretičnem delu predstavljamo pojme, kot so gospodinja, kmečko gospodinjstvo ter kmečka družina, kjer smo opisali tudi tipologijo kmečkih družin.	In the theoretical part, I have presented the following concepts: ″housewife″, ″rural household″ and ″rural family″. In addition, I have described the typology of rural families.	Yes
1.249	V nadaljevanju smo predstavili tradicionalno dojemanje kmečkih žensk, njihovo obravnavo skozi čas v slovenski literaturi, pojasnili smo procese, ki so vplivali na spremembo položaja kmečkih žensk skozi zgodovino ter se osredotočili na delo kmečkih žensk (delovni dan, delitev dela, vrednotenje dela).	I have explained the processes that have influenced the change in the situation of rural women through history and focused on their work (working day, divison of labour, work evaluation). Furthermore, I have shed light on the traditional perception of peasant women and their treatment over time in Slovene literature.	Yes
1.217	Ugotovili smo, da so tradicionalni vzorci delitve dela na kmetiji še vedno prisotni, saj smo iz analize anket in literature ugotovili, da ženske opravljajo večino del vezanih na dom in družino, to pa so gospodinjska dela in vzgoja otrok.	Hence, the majority of work related to home and family (housework and child-rearing) is performed by women. By analyzing the conducted survey and examining the literature, I have come to the conclusion that the division of farm labour more or less still follows traditional patterns.	Yes

[2] Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7, 597–610.

[3] Tomaž Erjavec, Darja Fišer, and Nikola Ljubešić. 2021. The KAS corpus of Slovenian academic writing. *Language Resources and Evaluation*, 55, 2, 551–583.

[4] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *arXiv preprint arXiv:2007.01852.*

[5] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. In *Internationmal Conference on Representation Learning*.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12, 2825–2830.

Indeks avtorjev / Author index