RASLAN 2021
Recent Advances in Slavonic Natural Language Processing

A. Horák, P. Rychlý, A. Rambousek (eds.)

RASLAN 2021

Recent Advances in Slavonic Natural Language Processing

Fifteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2021
Karlova Studánka, Czech Republic,
December 10–12, 2021
Proceedings

Tribun EU
2021

Proceedings Editors

Aleš Horák
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: `hales@fi.muni.cz`

Pavel Rychlý
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: `pary@fi.muni.cz`

Adam Rambousek
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: `rambousek@fi.muni.cz`

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the Czech Copyright Law, in its current version, and permission for use must always be obtained from Tribun EU. Violations are liable for prosecution under the Czech Copyright Law.

Editors © Aleš Horák, 2021; Pavel Rychlý, 2021; Adam Rambousek, 2021
Typography © Adam Rambousek, 2021
Cover © Petr Sojka, 2010
This edition © Tribun EU, Brno, 2021

ISBN 978-80-263-1670-1
ISSN 2336-4289

Preface

This volume contains the Proceedings of the Fifteenth Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2021) , organized by the NLP Consulting, s.r.o. and held on December 10$^{\text{th}}$–12$^{\text{th}}$ 2021 in Karlova Studánka, Sporthotel Kurzovní, Jeseníky, Czech Republic.

The RASLAN Workshop is an event dedicated to the exchange of information between research teams working on the projects of computer processing of Slavonic languages and related areas going on in the NLP Centre at the Faculty of Informatics, Masaryk University, Brno. RASLAN is focused on theoretical as well as technical aspects of the project work, on presentations of verified methods together with descriptions of development trends. The workshop also serves as a place for discussion about new ideas. The intention is to have it as a forum for presentation and discussion of the latest developments in the field of language engineering, especially for undergraduates and postgraduates affiliated to the NLP Centre at FI MU.

Topics of the Workshop cover a wide range of subfields from the area of artificial intelligence and natural language processing including (but not limited to):

* text corpora and tagging
* syntactic parsing
* sense disambiguation
* machine translation, computer lexicography
* semantic networks and ontologies
* semantic web
* knowledge representation
* logical analysis of natural language
* applied systems and software for NLP

RASLAN 2021 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 19 papers were accepted, contributed altogether by 31 authors. Our thanks go to the Program Committee members and we would also like to express our appreciation to all the members of the Organizing Committee for their tireless efforts in organizing the Workshop and ensuring its smooth running. In particular, we would like to mention the work of Aleš Horák, Pavel Rychlý and Marek Hríbik. The TEXpertise of Adam Rambousek (based on LATEX macros prepared by Petr Sojka) resulted in the extremely speedy and efficient production of the volume which you are now holding in your hands. Last but not least, the cooperation of Tribun EU as a publisher and printer of these proceedings is gratefully acknowledged.

Brno, December 2021 Karel Pala

Table of Contents

I NLP Applications

New Technology Platform for the Multilingual Sign Language Dictionary 3
Adam Rambousek

Application of Super-Resolution Models in Optical Character Recognition of Czech Medieval Texts 11
Mikuláš Bankovič, Vít Novotný, and Petr Sojka

Detecting Online Risks and Supportive Interaction in Instant Messenger Conversations using Czech Transformers 19
Ondřej Sotolář, Jaromír Plhák, Michal Tkaczyk, Michaela Lebedíková, and David Šmahel

When Tesseract Brings Friends: Layout Analysis, Language Identification, and Super-Resolution in the Optical Character Recognition of Medieval Texts 29
Vít Novotný, Kristýna Seidlová, Tereza Vrabcová, and Aleš Horák

Precomputed Word Embeddings for 15+ Languages 41
Ondřej Herman

II Semantics and Language Modelling

Using FCA and Concept Explications for Finding an Appropriate Concept 49
Marek Menšík, Adam Albert, and Tomáš Michalovský

Evaluating Long Contexts in the Czech Answer Selection Task 61
Marek Medveď, Radoslav Sabol, and Aleš Horák

Questions and Answers on Dynamic Activities of Agents 71
Marie Duží

Conceptual Framework for Process Ontology 83
Martina Číhalová

Towards Domain Robustness of Neural Language Models 91
Michal Štefánik, Petr Sojka

III Morphology and Syntax

Approaching Punctuation Errors in the New Proofreader of Czech 107
Vojtěch Mrkývka

Evaluating the State-of-the-Art Sentence Alignment System on Literary Texts 115
Edoardo Signoroni

Building a Dataset for Detection of Verb Coordinations with a Shared Argument 125
Helena Medková

DMoG: A Data-Based Morphological Guesser 135
Vojtěch Kovář, Pavel Rychlý

IV Text Corpora

When Word Pairs Matter 141
Michaela Denisová, Pavel Rychlý

Transferability of General Polish NER to Electronic Health Records 151
Krištof Anetta and Mahmut Arslan

A Case Study of High-Frequency Dictionary Collocations in a Spoken Corpus 161
Maria Khokhlova

Website Properties in Relation to the Quality of Text Extracted for Web Corpora 167
Vít Suchomel, Jan Kraus

Development of HAMOD: a High Agreement Multi-lingual Outlier Detection dataset 177
Miloš Jakubíček, Emma Romani, Pavel Rychlý, and Ondřej Herman

Subject Index 185

Author Index 187

Part I

NLP Applications

New Technology Platform for the Multilingual Sign Language Dictionary

Adam Rambousek

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech republic
`rambousek@fi.muni.cz`

Abstract. Since 2014, Teiresiás Centre at Masaryk University is co-ordinating the project to create the multilingual sign language dictionary. Natural Language Processing Centre is developing the editing and browsing web application for the dictionary. Originally, the application was based on the DEB dictionary platform with Sedna XML database for data storage. In course of the project, more languages were added, entry structure is more complex, larger teams from several countries are working on the dictionary creation, and website design was not working very well with modern web browsers. We realized that in order to increase the response speed of the application we need to refactor the whole technology platform. In 2020 and 2021, completely new application was designed and developed. This paper this describes the overall structure of the platform, technologies used to build the application and the process of data migration to the new database system.

Keywords: Dictionary editing · Dictionary writing system · Sign language · XML · JSON · MongoDB database

1 Introduction

In 2014, the Teiresiás Centre at Masaryk University was co-ordinating the project which aimed to build the Czech Sign Language dictionary connected with the Czech dictionary. Several organizations were working on the dictionary data, and the Natural Language Processing Centre was asked to develop web application to view and edit dictionary entries. Application was built using the DEB platform tools [9,5] – data were encoded in the XML format and stored in the Sedna XML database, for editing custom web editor was developed in Javascript, for viewing entries were converted from the XML format to HTML using XSLT templates. More details about the application are described in [8].

1.1 Languages and Entry Structure

Over the years, more international organizations joined the project and thus more languages were added. Dictionary application is called *Dictio – Multilin-*

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 3–10, 2021. © Tribun EU 2021

gual dictionary focused on sign languages[1]. Currently, the dictionary contains the following languages:

- Czech Sign Language (Český znakový jazyk, ČZJ),
- Slovak Sign Language (Slovenský posunkový jazyk, SPJ),
- Austrian Sign Language (Österreichische Gebärdensprache, ÖGS),
- American Sign Language (ASL),
- International Signs (IS),
- Czech,
- Slovak,
- German,
- English.

General entry structure is the same for all languages, however level of details in each part is different for various languages:

- headword,
- grammar information (at least Part-of-Speech, ideally all morphological details),
- etymology of the word or sign,
- stylistic information (regional or limited usage, etc.),
- for sign languages, transcription into SignWriting or HamNoSys [10,7],
- meanings
    - definition,
    - usage examples,
    - translations,
    - other semantic relations (e.g. synonyms, hypernyms).

Of course, the main difference between sign and spoken languages is the headword representation – headword is represented with the video recordings (front and side view) of the person showing the sign. In Dictio, unlike in other sign language dictionaries, even the definition and usage examples are presented as video recordings in sign language.

As for translations, at least the entries in sign language and its spoken counterpart (e.g. ČZJ and Czech, or ÖGS and German) are connected. But it is possible to add translations to any other language and web application supports searching in all of the language pairs.

Currently, Dictio contains 158,357 entries and 70,501 videos altogether, see Table 1 for details about the number of entries and recordings in each language.

2 Technology

After evaluation of tools used in the first version of Dictio, it was decided to implement most parts of the application from scratch.

---

[1] Available at `https://www.dictio.info/`.

Table 1: Number of entries and video recordings per language

language	entries	videos
Czech	120,274	
Czech Sign Language	12,526	44,330
German	5,652	
Slovak	5,590	
English	5,555	
Slovak Sign Language	4,812	17,300
Austrian Sign Language	3,436	7,400
International Signs	369	1,050
American Sign Language	127	290

Main part of the application logic was implemented in Ruby programming language[2] [3]. Some complex underlying functions may be kept with just a small updates, e.g. combining the SignWriting signs for collocation entries, or processing inter-language relations changes. For that reason, we decided to implement new application in Ruby, but updating the code from Ruby 1.8 to Ruby 2.6. Apart from keeping with current development, this update also introduced better handling of UTF-8 strings. Thus, all the tools and libraries used in the new application need to support Ruby.

2.1 Database

Entry structure is very complex and while it is stable after the development of the first version, there might still be structure changes in the future. Originally, entries were saved in the XML format and stored in Sedna XML database [4]. We needed to either keep the XML format, or use format with the same complexity.

With growing number of entries and links between them, the performance of Sedna XML database was getting worse. Unfortunately, Sedna is no longer actively developed, thus we had to select another database. We evaluated performance benchmarks for open-source XML and NoSQL databases. We decided to use MongoDB NoSQL database[3] [1,6].

MongoDB stores documents in the JSON format [2], or more specifically in the BSON ("Binary JSON") format[4]. BSON format is a binary representation of the JSON documents with support for more complex data types, and was designed to be more efficient both for the storage space, and the reading speed.

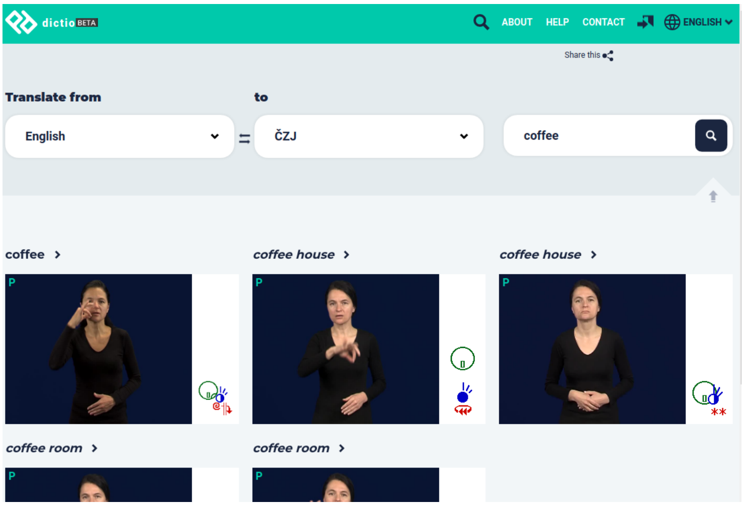
Because of the document format change, all the entries and metadata in the database had to be converted. This also proved to be good opportunity to clean the entry structure. We removed unnecessary nesting of data where possible to make the structure more readable. In the Sedna database, some values were

---

[2] `https://www.ruby-lang.org/`

[3] `https://www.mongodb.com/try/download/community`

[4] BSON format specification is available at `https://bsonspec.org/`. JSON and BSON formats are compared at `https://www.mongodb.com/json-and-bson`.



Translate from English

to ČZJ

coffee

Fig. 1: Translations from English to Czech Sign language.

duplicated (e.g. information about the target entry of translation link) to speed up querying and displaying. This is not needed anymore and each information is stored only once. Originally, each language had separate database collection for entries and for video metadata. In MongoDB, all entries are stored in single database with additional language attribute (similarly for video metadata).

On the backend side, no big changes were needed, because even in the first version all the XML data were converted to objects before using them in the application. This is much easier with BSON documents provided by the MongoDB API.

On the frontend side, the editor for creating and updating the entries had to be updated. The application is implemented in JavaScript and provides complex editing form for users. Fortunately, we had to update just the two functions: to load the XML document from the database and parse the data to form boxes, and to get the form data and send the XML document to the database. Obviously, these functions were re-implemented to work with the documents in JSON format.

2.2 Web Application Tools

Original version of Dictio used the Webrick server to process network requests and a set of custom templates and XSL stylesheets to display the web pages. Main disadvantages of the Webrick server are the worse performance with high load of requests, and support for only single-threaded processing.

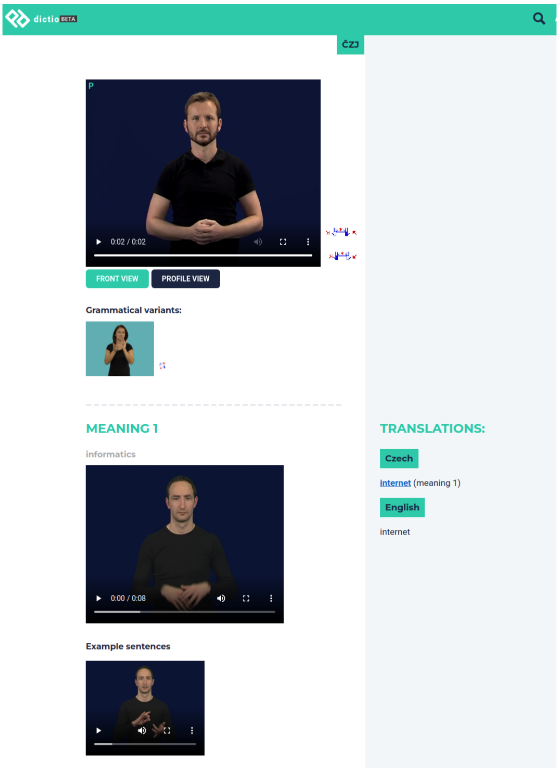

dictio BETA
ČZJ
0:02 / 0:02
FRONT VIEW
PROFILE VIEW
Grammatical variants:
MEANING 1
informatics
0:00 / 0:08
Example sentences
TRANSLATIONS:
Czech
internet (meaning 1)
English
internet

Fig. 2: Full details of single Czech Sign Language entry.

After performance evaluation of existing tools, we decided to use the Sinatra framework[5] for creating web applications. Sinatra is used for request processing, routing, user authentication, user session setting and application interface.

To display web pages with the data to the users, we selected the Slim template engine[6] Templates in Slim contain as little HTML formatting as possible, document structure is based on template indentation, and main focus in template writing is on the data. It is also possible to re-use and combine templates, which is advantage for well arranged implementation. Completely new web page design was created with support for mobile devices. See Figure 1 for example of result for translation search from English to Czech Sign Language. Figure 2 shows an example with full information about single entry in Czech Sign Language. See Figure 3 for example of layout for mobile devices, with results of translation from English to Czech Sign Language.

---

[5] Available at `http://sinatrarb.com/`.

[6] Available at `http://slim-lang.com/`.

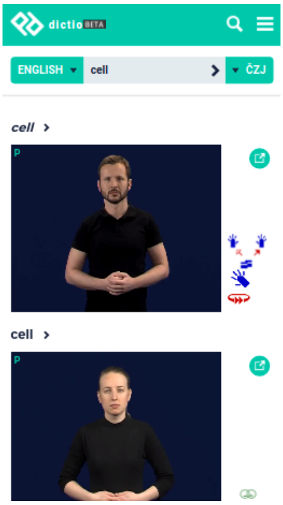


Fig. 3: Responsive design for mobile devices.

3 Platform Structure

In the Dictio project, there are several groups of users of the application with various needs:

- general public – browsing and querying the entries,
- editors – adding or updating the entries, uploading video recording, based on the department they belong to,
- dictionary managers – reviewing entries, assigning work based on reports about missing parts of entries, managing users and their access permissions.

In the original Dictio version, all users were working on the same server. Also the database and all the video files were stored on the same machine. This arrangement had bad impact on the overall performance and user experience. For example, when many users were browsing the dictionary, the entry editing application was responding slower. Similarly, when mass import of video recording was under way, users were waiting too long for entry display.

To improve the application performance and also to keep different tasks separate, we designed new platform structure. Application is now split into five independent virtual servers, provided by the MetaCentrum Cloud[7]:

- database server with MongoDB,
- file server with all the video recordings (`files.dictio.info`),
- public viewing server (`www.dictio.info`),
- editing server (`edit.dictio.info`),
- administration server (`admin.dictio.info`).

---

[7] `https://cloud.muni.cz/`

All three web servers (www, edit, admin) share the same source code and thanks to Sinatra conditional routing only the appropriate parts and templates are provided. Database server is accessible only via internal network from the web servers, and is not open to public network.

4 Conclusion and Future Developments

We re-implemented the Dictio multilingual sign language dictionary as completely new web application. We decided to change the database, document storage format, web framework, and template engine. Using current technology and more modular application structure is providing better performance and better experience for users. Currently, all functionality of the original application is supported. New application is in regular use since March 2021 and we are continuously adding new features based on user feedback.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Chodorow, K., Dirolf, M.: MongoDB: The Definitive Guide. O'Reilly Media, Inc., 1st edn. (2010)
2. Crockford, D.: JSON, The Fat-Free Alternative to XML. In: Proceedings of XML 2006. Boston, USA (2006), `http://www.json.org/xml.html`
3. Flanagan, D., Matsumoto, Y.: The Ruby Programming Language. O'Reilly, first edn. (2008)
4. Fomichev, A., Grinev, M., Kuznetsov, S.: Sedna: A Native XML DBMS. Lecture Notes in Computer Science **3831**, 272 (2006)
5. Horák, A., Rambousek, A.: Lexicographic tools to build new encyclopaedia of the czech language. The Prague Bulletin of Mathematical Linguistics **2016** (2016). https://doi.org/http://dx.doi.org/10.1515/pralin-2016-0019, `https://ufal.mff.cuni.cz/pbml/106/art-horak-rambousek.pdf`
6. Jose, B., Abraham, S.: Performance analysis of nosql and relational databases with mongodb and mysql. Materials Today: Proceedings **24**, 2036–2043 (2020). https://doi.org/https://doi.org/10.1016/j.matpr.2020.03.634, `https://www.sciencedirect.com/science/article/pii/S2214785320324159`
7. Kato, M.: A Study of Notation and Sign Writing Systems for the Deaf. Intercultural Communication Studies **17**(4), 97–114 (2008)
8. Rambousek, A., Horák, A.: Management and Publishing of Multimedia Dictionary of the Czech Sign Language. In: Biemann, C., Handschuh, S., Freitas, A., Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems, NLDB 2015. pp. 399–403. Lecture Notes in Computer Science, Springer (2015). https://doi.org/10.1007/978-3-319-19581-0_37

9. Rambousek, A., Horák, A., Parkin, H.: Software tools for big data resources in family names dictionaries. Names **66** (2018). https://doi.org/http://dx.doi.org/10.1080/00277738.2018.1453276, `https://www.tandfonline.com/doi/full/10.1080/00277738.2018.1453276`
10. Sutton, V.: SignWriting Basics. Center for Sutton Movement Writing, Incorporated (2009)

Application of Super-Resolution Models in Optical Character Recognition of Czech Medieval Texts

Mikuláš Bankovič, Vít Novotný, and Petr Sojka

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`{456421,witiko}@mail.muni.cz, sojka@fi.muni.cz`
https://mir.fi.muni.cz/

Abstract. Optical character recognition (OCR) of scanned images is used in multiple applications in numerous domains and several frameworks and OCR algorithms are publicly available. However, some domains such as medieval texts suffer from low accuracy, mainly due to low resources and poor quality data. For such domains, preprocessing techniques help to increase the accuracy of OCR algorithms.

In this paper, we experiment with two super-resolution models: Waifu2x and SRGAN. We use the models to reduce noise and increase the image resolution of scanned medieval texts. We evaluate the models on the AHISTO project dataset and compare them against several baselines. We show that our models produce improvements in OCR accuracy.

Keywords: Super-resolution · Optical character recognition · Medieval texts

1 Introduction

The aim of the AHISTO project is to make documents from the Hussite era (1419–1436) available to the general public through a web-hosted searchable database. Although scanned images of letterpress reprints from the 19th and 20th century are available, accurate optical character recognition (OCR) algorithms are required to extract searchable text from the scanned images. However, the scanned images are noisy and low-resolution, which decreases OCR accuracy.

In our work, we develop image super-resolution models and data augmentation techniques for training these models. We use our image super-resolution models to increase the resolution of scanned pages and we evaluate the impact on the OCR accuracy on medieval texts.

In Section 2, we describe the related work in image super-resolution and the OCR of medieval texts. In Section 3, we describe our training and test datasets, data augmentation techniques, image super-resolution models, and baselines. In Section 4, we discuss the results of our evaluation. We conclude in Section 5 and offer directions for future work.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 11–18, 2021. © Tribun EU 2021

2 Related Work

Traditionally, the image processing techniques that improve the accuracy for OCR of medieval texts documents were primarily rule-based [3]. However, there has been a growing interest in using deep learning methods for OCR preprocessing.

In this section, we will present the recent work in deep learning methods for OCR preprocessing and in the OCR of medieval texts.

2.1 Super-Resolution Models

Walha et al. (2012) [17] showed that image super-resolution models based on learned dictionaries between low-resolution and high-resolution sparsely encoded patches improved performance on image upscaling. However, the computation demands for this algorithm were high. Nayef et al. (2014) [8] proposed selective patch processing, performing costly operations only on high variance patches and using bicubic interpolation otherwise.

As in many other domains, deep learning models that used convolution neural networks (CNNS) surpassed previous techniques for image super-resolution. These models included SRCNN [1] and more complex generative adversarial networks (GANS) such as SRGAN [6]. Nakao et al. (2019) [7] adapted the SRCNN loss function for text by maintaining sharp boundaries between letters.

Lat and Jawahar (2018) [5] used SRGAN to improve OCR accuracy. Su et al. (2019) [15] showed that adding ℓ_1 loss to the SRGAN model helps maintain detail in letterforms. Nguyen et al. (2019) [9] translated poorly visible letters to binarised letters using a variation of SRGAN and a weakly coupled dataset.

Fu et al. (2019) [2] suggested using cascaded networks consisting of CNN, improving OCR accuracy over both SRCNN and SRGAN. Ray et al. (2019) [12] and Randika et al. (2021) [11] added the gradient loss of the OCR algorithm to the image super-resolution model, creating an end-to-end deep learning framework.

2.2 Optical Character Recognition Engines

In 2020, the second author [10] showed that Tesseract 4 [4] gave the best trade-off between speed and OCR accuracy for medieval texts. Therefore, we only experiment with Tesseract 4 in our work.

3 Methods

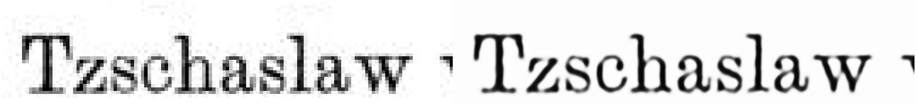
In this section, we discuss our training and test datasets, the data augmentations we used, and our super-resolution models and baselines.

3.1 Datasets

As our training dataset for the image super-resolution models, we used a born-digital PDF version of the sixth tome of the book *Codex Diplomaticus et Epistolaris*

Regni Bohemiae [16], which contains a collection of medieval texts (1278–1283) from the Kingdom of Bohemia.

As our test dataset for the OCR accuracy, we used the AHISTO dataset. The dataset contains 65,348 pairs of low-resolution and high-resolution scanned images [10, Section 3.1], see Figure 1. For 120 scanned images, the dataset contains human annotations with correct OCR texts. We used the human annotations with the word error rate (WER) measure [14] to evaluate the OCR accuracy. See another article from these proceedings on page 29 for more information about the human annotations and the WER evaluation measure.



(a) Low resolution image patch (b) High resolution image patch

Fig. 1: Low-resolution and high-resolution image patches from our test dataset

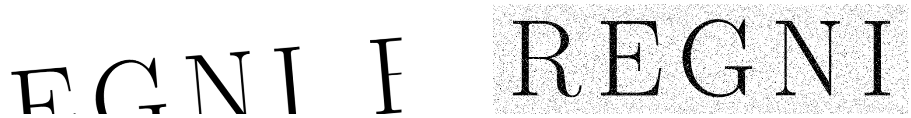
3.2 Data Augmentations

We augmented images as shown in Figure 2 with the following methods:

- *Rotate* rotates by an angle, blank spaces are filled using bicubic interpolation.
- *JPEG noise* recompresses image to JPEG quality.
- *Salt and pepper* adds random black and white pixels to the image.

REGNI
REGNI

(a) Original image (b) Image augmented with JPEG noise



(c) Rotated image (d) Salted and peppered image

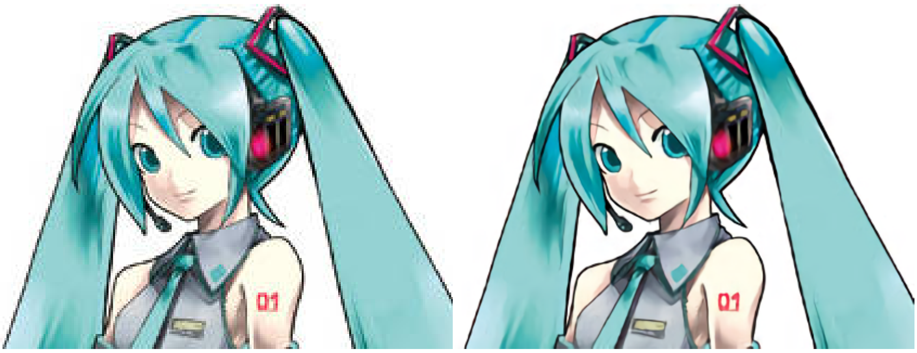
Fig. 2: Data augmentations of low-resolution images

3.3 Super-Resolution Models

For image super-resolution, we use the SRGAN and SRCNN models.

SRGAN has multiple hyperparameters to optimise: the number of epochs, the learning rate, and the size of the image patches. We augment SRGAN to work with greyscale weights, reducing the number of parameters approximately by a factor of 3. We also experiment with removing the discriminator part of an SRGAN network (further known as SRRESNET) [6]. Our code is available online.[1]

Due to time constraints, we do not train our own SRCNN model. Instead, we use public models[2] (*Waifu2x*) pre-trained on drawn images,[3] see Figure 3.



(a) Low resolution image (b) High resolution image

Fig. 3: Low-resolution and high-resolution images from the training dataset of pre-trained Waifu2x models. The image is licensed under CC BY-NC by piapro.

3.4 Baselines

As our baselines, we used the original low-resolution and high-resolution image pairs. Additionally, we also used the bilinear interpolation and the Potrace [13] rule-based image vectorizer to upscale the low-resolution images.

4 Results

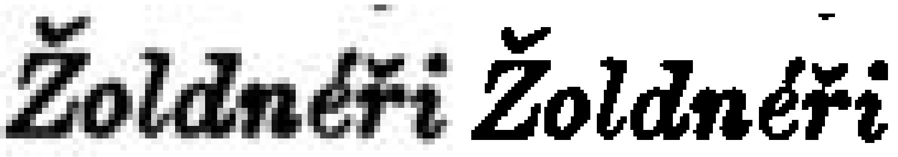
Table 1 shows that high-resolution images have better performance than low-resolution images. Specific settings performed even better than original high-resolution images, which is unexpected in the case of bilinear interpolation baseline. Waifu2x with added JPEG noise achieved the best performance.

---

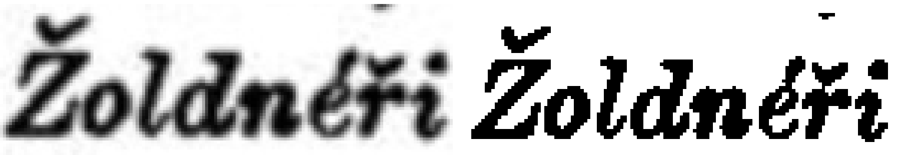
[1] https://github.com/xbankov/Fast-SRGAN

[2] https://github.com/nagadomi/waifu2x/tree/master/models/cunet/art

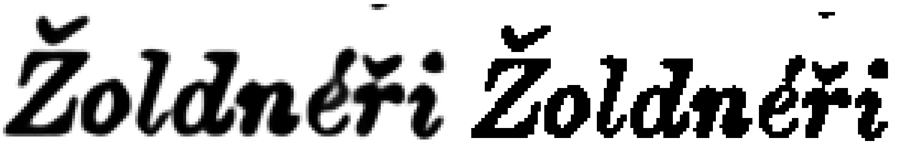
[3] https://github.com/nagadomi/waifu2x/issues/263


Žoldněři
Žoldněři

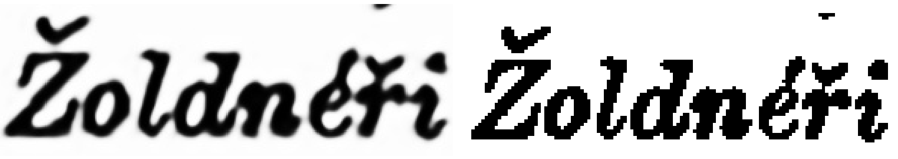
(a) Low-resolution image vs. high-resolution image


Žoldněři
Žoldněři

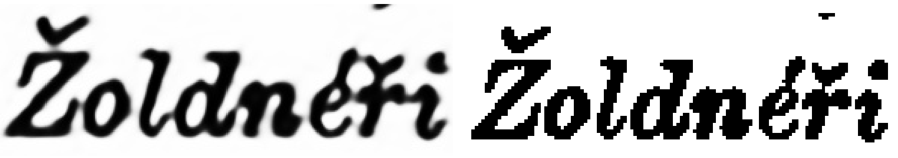
(b) Bilinear interpolation image vs. high-resolution image


Žoldněři
Žoldněři

(c) Waifu2x image vs. high-resolution image


Žoldněři
Žoldněři

(d) SRRESNET without any modifications vs. high-resolution image


Žoldněři
Žoldněři

(e) SRRESNET image rotated by angle 2° vs. high-resolution image

Fig. 4: The low resolution image in Fig. 4a is an input to other methods. In each figure in the left is tested example and on the right of each figure is the same original high resolution scan.

Table 1: Impact of super-resolution on OCR accuracy. Best results are bold.

Architecture	Modification	Epochs	WER (%)
Low-resolution			14.75
Bilinear			7.77
Potrace			9.29
High-resolution			8.74
SRGAN		20 + 1	9.63
SRRESNET		20	8.95
SRRESNET	binarize	20	9.72
SRRESNET	grayscale	20	8.79
SRRESNET	rotate 2°	20	8.19
SRRESNET	rotate 2° + greyscale	20	8.32
Waifu2x			7.46
Waifu2x	JPEG noise		**7.45**

We observed that SRRESNET bested SRGAN in every setting. Therefore, we only list a single result for SRGAN in Table 1 for comparison. The grayscale variant performs comparably with RGB. Most of the augmentations did not perform well, either due to wrong parameters or inappropriate design.

SRRESNET in Figure 4d looks intuitively better than bilinear interpolation in Fig. 4b. It is unclear why bilinear performs better within the Tesseract framework. In Figure 4c created by Waifu2x, the letters are separated and smoothed. Therefore the best result in OCR performance is justified. In contrast, in 4d, the letters *ř* and *i* are connected and can mislead the OCR engine.

5 Conclusion and Future Work

In our work, we have experimented with data augmentation methods for SRGAN. We tested the impact of super-resolution models on the OCR of medieval texts. We concluded that the resolution of the image matters for the Tesseract OCR engine. Even bilinear interpolated images work significantly better than original low-resolution images.

We also showed that the grayscaling of weights can be used to decrease the size and training time of image super-resolution models without any adverse effect on OCR accuracy.

The victory of the Waifu2x models, which were pre-trained on data from different domain, shows that the size of our training dataset was insufficient for training larger models such as SRGAN and SRRESNET. Future work should collect more training data, for example by typesetting the OCR texts produced from scanned document pages.

We realised that our salt and pepper augmentation did not reflect real scan damage. Future work should focus at more realistic *damaged scan* augmentations, such as modified salt and pepper resembling ink droplets and blank spots after ink has peeled off the paper and flaked away.

Future work should also experiment with modified loss functions that improve the performance of image super-resolution techniques with text.

Acknowledgements. The South Moravian Centre graciously funded the second author's work for International Mobility as a part of the Brno PhD. Talent project. The research was also supported by TAČR Éta, project number TL03000365.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 184–199 (2014)
2. Fu, Z., Kong, Y., Zheng, Y., Ye, H., Hu, W., Yang, J., He, L.: Cascaded detail-preserving networks for super-resolution of document images. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 240–245. IEEE Computer Society (2019). https://doi.org/10.1109/ICDAR.2019.00047
3. Jon M. Booth, J.G.: Optimizing OCR accuracy on older documents: A study of scan mode, file enhancement, and software product, https://www.govinfo.gov/media/WhitePaper-OptimizingOCRAccuracy.pdf, [cit. 2021-11-06]
4. Kay, A.: Tesseract: An Open-Source Optical Character Recognition Engine. Linux Journal **2007**(159), 2 (Jul 2007), https://dl.acm.org/doi/10.5555/1288165.1288167
5. Lat, A., Jawahar, C.V.: Enhancing OCR accuracy with super resolution. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3162–3167. IEEE (2018)
6. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017), https://arxiv.org/abs/1609.04802v5
7. Nakao, R., Iwana, B.K., Uchida, S.: Selective super-resolution for scene text images. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 401–406 (2019). https://doi.org/10.1109/ICDAR.2019.00071
8. Nayef, N., Chazalon, J., Gomez-Krämer, P., Ogier, J.M.: Efficient example-based super-resolution of single text images based on selective patch processing. In: 2014 11th IAPR International Workshop on Document Analysis Systems. pp. 227–231 (2014). https://doi.org/10.1109/DAS.2014.25
9. Nguyen, K.C., Nguyen, C.T., Hotta, S., Nakagawa, M.: A character attention generative adversarial network for degraded historical document restoration. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 420–425 (2019). https://doi.org/10.1109/ICDAR.2019.00074
10. Novotný, V.: When Tesseract Does It Alone. In: Proceedings of the Fourteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2020. pp. 3–12 (2020), https://nlp.fi.muni.cz/raslan/2020/paper1.pdf
11. Randika, A., Ray, N., Xiao, X., Latimer, A.: Unknown-box approximation to improve optical character recognition performance, https://arxiv.org/abs/2105.07983v1, [cit. 2021-11-03]

12. Ray, A., Sharma, M., Upadhyay, A., Makwana, M., Chaudhury, S., Trivedi, A., Singh, A., Saini, A.: An end-to-end trainable framework for joint optimization of document enhancement and recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 59–64 (2019). https://doi.org/10.1109/ICDAR.2019.00019
13. Selinger, P.: Potrace: a polygon-based tracing algorithm (2003), http://potrace.sourceforge.net/potrace.pdf, [cit. 2021-11-07]
14. Soukoreff, R.W., MacKenzie, I.S.: Measuring errors in text entry tasks: An application of the levenshtein string distance statistic. In: CHI'01 extended abstracts on Human factors in computing systems. pp. 319–320 (2001)
15. Su, X., Xu, H., Kang, Y., Hao, X., Gao, G., Zhang, Y.: Improving text image resolution using a deep generative adversarial network for optical character recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1193–1199 (2019). https://doi.org/10.1109/ICDAR.2019.00193
16. Sviták, Z., Krmíčková, H., Krejčíková, J., Friedrich, G.: Codex diplomaticus et epistolaris Regni Bohemiae. Tomi VI, fasciculus primus, Inde ab A. MCCLXXVIII usque ad A. MCCLXXXIII. Academia (2006)
17. Walha, R., Drira, F., Lebourgeois, F., Alimi, A.M.: Super-resolution of single text image by sparse representation. In: Proceeding of the Workshop on Document Analysis and Recognition. p. 22–29 (2012). https://doi.org/10.1145/2432553.2432558

Detecting Online Risks and Supportive Interaction in Instant Messenger Conversations using Czech Transformers

Ondřej Sotolář, Jaromír Plhák, Michal Tkaczyk, Michaela Lebedíková, and
David Šmahel

IRTIS, Faculty of Informatics, Masaryk University,
Botanicka 68a, 602 00, Brno, Czech Republic
`{xsotolar,xplhak}@fi.muni.cz`, `tkaczyk@fss.muni.cz`,
`{m.lebedikova,davs}@mail.muni.cz`

Abstract. We present a comparison of state-of-the-art models for text classification of Online Risks and Supportive Interaction in anonymized Instant Messenger conversations held in Czech. We compare the transformer models Czert, RobeCzech, and FERNET-C5 with the Fasttext classifier as a baseline. For the comparison, we build a novel dataset with five subcategories for the Online Risks and five for the Supportive Interaction. We solve the balanced classification problem achieving 75.44 - 89.66 F1 score depending on the category. Our results show that the transformer models perform consistently better than the baseline.

Keywords: Online Risks · Supportive Interaction · Facebook Messenger · Text Classification

1 Introduction

Starting Natural Language Processing research in a new language domain brings uncertainty about how existing models and tools will perform in it. In such case, it is a good practice to compare several candidate models and select the best-performing ones to develop further.

In our case, the domain of interest is composed of anonymized Instant Messenger (IM) conversations of Czech adolescents conducted in Czech. Current research [1] is trying to examine the effect of smartphone use on the well-being of adolescents through analyzing data collected on-device. The IM conversations constitute a significant portion of this data, and the classification will allow for the measurement of smartphone use with high validity. It will provide insights into what the users actually do on their devices in IM conversations and what is the possible impact on their well-being.

So far, this specific domain has been under-researched in NLP. We try to establish the difficulty of classifying the IM messages (without context) into the respective sub-categories of the Online Risks and Supportive Interaction categories, described in 3.2. We perform a model comparison by fine-tuning four new Czech transformer models using the Fasttext classifier as the baseline.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 19–28, 2021. © Tribun EU 2021

2 Related work

The work in the domain of text obtained from social networks is highly diverse, as it is an active area of research. Close in the language domain and study participants' age is the BlackBerry project [27] that examined adolescents' text messages. The authors of [26] classify social network messages of adolescents from various sources by ensembling various statistical machine learning models trained on word N-grams. Both of the mentioned works were carried out on English corpora. In Czech, sentiment analysis was carried out on a dataset of Czech Facebook posts in [9]. All of these works use methods pre-dating the widespread use of text embeddings.

Contemporary NLP classification methods leverage the strength of pre-trained deep language representation models, which have surpassed statistical approaches, such as used in [15] in various Text Classification (TC) tasks. A systematic review of the Neural Network (NN) architectures in [17] gave us guidance on the choice of the baseline model. We chose those transformer models that achieve SOTA for Czech, based on the comparison in [14]. Until very recently, the multilingual models SlavicBERT [2], mBERT [7], and XLM-RoBERTa-base [6] achieved SOTA results for Czech. They were recently surpassed in classification by BERT-based models Czert-B [22], FERNET-C5 [14], and RoBERTa based RobeCzech [25] that achieve comparable results with larger multilingual models, such as the XLM-RoBERTa-large [6]. Czert and RobeCzech are trained on a combination of Czech National Corpus [12], Czech Wikipedia dump, and Czech news crawl. FERNET is trained on C5, a new Common Crawl-based corpus. For completeness, we also measured the smaller ELECTRA [4] model, Small-E-Czech [21], trained on a Czech web crawl and search queries.

3 Methods

3.1 Language Domain

The domain of private IM conversation is much less explored than the domain of publicly available text gathered from social networks. Arguably, because such data are hard to obtain, they may contain sensitive information and thus need to be anonymized, which is challenging. To solve it, we used the methods described in [24]. Some features and issues in this domain are given by the fact that the communication is held in Czech, it is conducted in private, through IM communication tools, and it is communication between adolescents, their peers, and sometimes also caregivers, such as parents. The dialogues are commonly conducted in informal language. Their syntactic, stylistic, and grammatical quality is considerably lower than formal styles, such as the encyclopedic and journalistic styles, predominantly represented in the pre-trained models' training corpora. The difference from the informal but for-public intended text, such as status messages from social networks, forums, discussions, and chat room conversations, all of which also occur in the training sets of language models, remains un-quantified. Ultimately, when using such language models on tasks in our particular domain, the data cannot be considered to be within-domain [8].

3.2 Annotated Corpus

We have created an annotated corpus of Facebook Messenger conversations of adolescents participating in our study (*N=17, 13-17 years old*). Out of all collected conversation records, we drew stratified batches of conversation samples of representative size to ensure variability of the phenomena under consideration in the annotated corpora. The total size of the annotated corpora for SI and OR, expressed in number of rows of text, is (*N=270,760, N=196,196*), also shown in Table 2 among other statistics.

The **Annotation categories**, i.e. Supportive Interactions (SI) and Online Risks (OR) were derived from relevant research and theory in the fields of psychology and communication [18,5,3,23,20]. Both categories refer to different, conceptually unrelated types of communicative behavior, and they differ from each other also in terms of their linguistic features. SI covers a range of communicative behaviors oriented at achieving the same intention, which is providing social support through interpersonal communication to another participant of conversation. Data falling under the OR category are defined by the mere fact of referring to a particular topic, i.e., different types of risks to adolescents' health and development, no matter whether at the interactional or ideational level of language [10], e.g., it could be instances of online aggression directed at conversation participants but also references to aggressive behavior conducted by someone else offline.

Since each of the two categories contained several sub-categories (see Table 1), the annotation was posed as a multi-label problem for each category[1]. Labels could be assigned to either a *single row* of a conversation or a *block of consecutive rows*. In order to create contextual units for the annotators to evaluate rows or blocks, they were delimited by the conversation turns of chat participants.

We used Cohen's κ [13] to measure the **IAA** because each category has been annotated by two annotators (see Table 2 for the achieved IAA). In the case of SI, positive examples were frequent enough, and we achieved a satisfactory level of IAA. It oscillated between batches between moderate (.41 to .60) and substantial (.61 to .80) and was constantly improving. For OR, the occurrences were rarer; thus, we abandoned the random sampling of batches. Instead, we first draw samples that scored the highest with preliminary classifiers trained on the previously annotated data, which improved the yield. The IAA oscillated between slight (.21-40) and moderate (.41 to .60) and improved inconsistently.

To sum up, for each category, we have obtained labels of different quality. Especially in the case of OR, the reliability of the data is not entirely satisfactory.

---

[1] While the annotation problem was indeed multi-label, due to various constraints, the annotators always assigned only the most probable label and indicated that there could be more labels on the particular line, leaving it unfinished. This effectively makes the problem multi-class.

Table 1: Description of categories.

Category	**Description**
Supportive Interactions	
Information Support	provide useful knowledge and information
Emotional Support	express intimacy, caring, liking, empathy, or sympathy
Social Companionship	convey a sense of belonging, inclusivity, will to spend time together in leisure, recreational activ.
Appraisal	express acceptance, respect, validation, esteem
Instrumental Support	offer practical help or resources, assistance in getting necessary tasks done
Online Risks	
Aggression, Harassment, Hate	use of or reference to offensive language and slander to cause harm
Mental Health Problems	reference to long-standing MH problems: suicide, self-harm, depression, eating disorders
Alcohol, Drugs	reference to experiences with alcohol and drugs
Weight Loss, Diets	discussions of weight-loss, working out and diets
Sexual Content	sexual or sexually suggestive discussions

Table 2: Statistics of the annotated corpus.

Category	**# rows labeled**	**P(cat)**	**κ**	**# blocks**
Supportive Interactions (N=270,760)				
Information Support	9967	5.08	0.685	5325
Emotional Support	9669	4.93	0.639	7284
Social Companionship	5317	2.71	0.599	4047
Appraisal	2338	1.19	0.65	1874
Instrumental Support	3331	1.7	0.604	2482
Online Risks (N=196,196)				
Aggression, Harassment, Hate	5382	1.99	0.47	3737
Mental Health Problems	3098	1.14	0.46	1605
Alcohol, Drugs	2288	1.17	0.609	1625
Weight Loss, Diets	91	0.03	-	46
Sexual Content	3563	1.32	0.485	2949

3.3 Training Dataset

The phenomena we are classifying are rare events (see column $P(cat)$ for the percentage of rows in Table 2). Solving the imbalance of a dataset that would respect the original distribution is not among the goals of this article; therefore, we built binarized balanced datasets. They are composed of all the positively labeled data per respective class, complemented by an equivalent amount of semi-randomly chosen negatively labeled data, in both cases labeled by at least one annotator. The positive labels often span across multiple rows of a single participant. We concatenated such cases into blocks (column *blocks* in Table 2) of one or more consecutive rows with the same label, thus reducing the overall example count. The negative examples were selected randomly but with paying attention to the distribution of several features of the positive blocks (character count, line count, number of participants), in an effort to minimize the statistical bias introduced by the undersampling. Preprocessing consisted only of lower-casing and removal of examples shorter than five characters.

3.4 Baseline Model

We chose the Fasttext classifier [11] as our baseline model, which is based on a shallow feed-forward NN using word embeddings as inputs. It can achieve high accuracy on many TC benchmarks, especially on datasets with high syntactic variance, which is our case. We have used the automatic tuning feature to determine ideal hyperparameters. We have also measured the impact of using pre-trained embeddings.

3.5 Transformer Models

BERT [7], is a transformer model pre-trained on a large corpus in a self-supervised fashion, with the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives. In MLM, the model randomly masks a portion (15%) of the words in the input, then inputs the sequence in the model and learns to predict the masked words. This is different from recurrent neural networks or from auto-regressive models like GPT [19] which mask the future tokens. In NSP, the model, given two sequences, learns to predict if the second sequence follows the first one. This way, the model learns low-level, bi-directional representations of the target language from which we can create a classifier by a process called fine-tuning. The model outputs a special token [CLS] that encodes the final hidden state h of the BERT model after inputting the sequence. Finally, a softmax layer is added on top of the model to predict the probability of label l:

$$p(l|\mathbf{h}) = softmax(W\mathbf{h}), \tag{1}$$

where W are the new layer's parameters which are learned by minimizing the cross-entropy loss using the task-specific dataset.

There are several variants of BERT that alter some of its components to either improve it, shrink it, or achieve some other goal. RoBERTa [16], whose goal is to improve the absolute performance, differs from BERT in the masking process, tokenization, and pre-training. In BERT, the masking is performed only once at data preparation time: the model masks each sequence a fixed number of times. Therefore, at training time, the model will only use those previously generated variations. On the other hand, in RoBERTa, the masking is done during training, each time a sequence is incorporated in a minibatch. As a result, the number of potentially different masked versions of each sequence is not bounded like in BERT. RoBERTa additionally uses a different style of BPE tokenization (same as GPT-2). While BERT highlights the merging of two subsequent tokens, RoBERTa's tokenizer instead highlights the start of a new token with a specific unicode character to avoid the use of whitespaces. Furthermore, RoBERTa removes the NSP task from pre-training. Thus, in theory, the RoBERTa model is more effectively regularized and can be trained for more epochs to achieve better results.

ELECTRA is a BERT-based architecture, whose goal is to shrink the network. Instead of using a masking token for the MLM, it provides plausible replacements sampled from a generator network. It offers solid performance while keeping the network several times smaller than BERT or RoBERTa.

4 Results

We summarize the experimental results in Table 3. They partially confirm the results of [14] by showing that the FERNET-C5 model performs among the two best models across our categories. However, in most experiments, RobeCzech could achieve comparable or better performance. The Czert model, being the first Czech transformer, is expectantly performing consistently worse than both the newer models. The performance of Small-E-Czech is also worse compared to the best models in all cases. On the other hand, the model is significantly smaller, and its training is faster. Surprisingly, the much simpler Fasttext model can approach the performance of the transformer models, provided that there is enough training data. On the categories with fewer training examples, the strength of transfer learning showed in the much larger gap in performance between the transformer models and Fasttext.

4.1 Hyperparameters

We optimized the hyperparameters globally for all transformer models and all categories at once. We based on the default values on [14] and used grid search only to slightly tweak them to fit our dataset and hardware. The results can be therefore easily compared with the previous works. The reported results use the following settings: (*batch size=128, peak learning rate=1e-5, warmup steps=1/3 no. total steps w. linear decay*). We trained for 20 epochs; however, with early stopping, which showed the ideal number of epochs be between 7-10, which confirms the 10 epoch setting used by [14].

For the Fasttext model, we used it's automatic hyperparameter optimization feature that resulted in (*dim=300, epoch=36, lr=0.05, lrUpdateRate=100, maxn=5, minn=2*) with other parameters left on default. Experiments with pretrained Fastext embeddings did not result in improvement.

Table 3: Results of binary classification. We report the cross-validated F1 score.

Category	Czert-B	RobeCzech	FERNET-C5	Small-E-Czech	Fasttext
Supportive Interactions					
Information Support	71.95	**75.44**	74.91	73.73	70.89
Emotional Support	74.63	76.67	**78.2**	72.94	74.05
Social Companionship	79.58	83.99	**84.74**	81.73	79.85
Appraisal	81.23	81.49	**85.87**	70.14	82.07
Instrumental Support	76.63	**82.12**	79.6	78.35	75.67
Online Risks					
Aggression, Harassment, Hate	84.41	**88.23**	**88.23**	83.24	83.24
Mental Health Problems	72.49	82.82	**85.11**	77.05	64.39
Alcohol, Drugs	87.17	**89.66**	87.6	81.40	63.22
Weight Loss, Diets	-	-	-	-	-
Sexual Content	70.62	74.33	**81.94**	67.72	63.16

4.2 Error Analysis

Many misclassified samples point to the obvious lack of context for each example. This causes the model to miss many finer points of the annotation manual, such as the instruction to assign a negative label to samples with a sarcastic connotation (sometimes expressed with an emoticon). However, including context would require a modification of the compared models, which is not among the goals of this article.

The analysis of high-certainty but misclassified predictions revealed that many samples rely on only one or two keywords, as shown in the model-view diagram of the *bertviz* tool [28] in Figure 1. If such keywords form a majority on one side of the binary classifier, it tends to classify all such samples into one class, some of them wrongly. Another reason for this class of error that we confirmed is that some of the misclassified samples are actually classified correctly, but the annotators disagreed on the label.

The analysis of low-certainty samples shows that these are, on average, considerably shorter than the high-certainty ones. They contain a number of one-word and text fragment samples which, in combination with the lack of context, does not provide the classifier enough input to perform well.

4.3 Discussion and Further Work

Our investigation yielded some interesting findings, such as the fact that the Fasttext model can rival the much larger transformers even without pretraining. While being a simpler model, the original implementation of the model is very efficient. That enables the search for hyperparameters to be several orders of magnitude faster than for the transformers models in the HuggingFace [29] library, which we used.

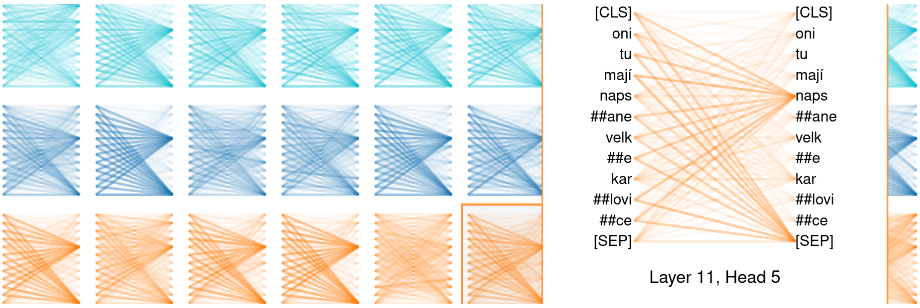


Fig. 1: Model-view of the last three layers of Czert for the high-certainty misclassified sequence, for the sub-category Information Support: *'oni tu mají napsane velke karlovice'*. The tokens on the left side of the bipartite graph provide attention to the right-side ones. We can see the repeating pattern on both the detail of the last layer's last head and the miniatures of other heads. The classifier is heavily biased towards the token *'naps'*, a part of the verb *'written'*, an expected keyword of this category.

Overall, we consider the results of this work to set solid baselines, to which new results can be compared. For further work, we suggest improving the regularization of the dataset by dropout or data augmentation, which could improve performance on the high-certainty misclassified samples by addressing the keyword issue. Additionally, further cleaning of the low-certainty samples could improve classification on this class of error. Furthermore, a more sophisticated hyperparameter search could improve the performance of the transformer models.

However, the obvious next step should be modeling the impact of the context of the messages. For example, [30] has shown that as far back as two months of previous dialogue can help improve the classification of new messages.

5 Conclusions

We have compared four new Czech transformer models on the task of text classification. We have shown that they provide a consistent improvement over the baseline Fasttext model and partially confirm the results from previous works, showing that the FERNET and RobeCzech models perform better than the Czert or Small-E-Czech models. In doing so, we prove that in the language domain of our dataset, i.e., short IM messages held in Czech, classification can be successfully performed even without the messages' context. We have built new annotated corpora for each of the sub-categories of Supportive Interactions and Online Risks categories, created datasets of them, and trained text classification models that have achieved 75.44 - 89.66 F1 score.

Acknowledgements. This work has received funding from the Czech Science Foundation, project no. 19-27828X.

References

1. Understanding the impact of technology on adolescent's well-being (FUTURE). `https://irtis.muni.cz/research/projects/future`, accessed: 2021-10-28
2. Arkhipov, M., Trofimova, M., Kuratov, Y., Sorokin, A.: Tuning multilingual transformers for language-specific named entity recognition. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing. pp. 89–93 (2019)
3. Bonino, S., Cattelino, E., Ciairano, S.: Adolescents and risk: Behaviors, functions and protective factors. Springer Milan (2005), `https://books.google.com.bn/books?id=FcFeNk_38-IC`
4. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
5. Cohen, S., Underwood, L.G., Gottlieb, B.H.: Social Support Measurement and InterventionA Guide for Health and Social Scientists: A Guide for Health and Social Scientists. Oxford University Press, New York, NY (09 2015). https://doi.org/10.1093/med:psych/9780195126709.001.0001
6. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Elsahar, H., Gallé, M.: To annotate or not? predicting performance drop under domain shift. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2163–2173 (2019)
9. Habernal, I., Ptáček, T., Steinberger, J.: Sentiment analysis in czech social media using supervised machine learning. In: Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis. pp. 65–74 (2013)
10. Halliday, M.A.K., Matthiessen, C.M.I.M.: An introduction to functional grammar / M.A.K. Halliday. Hodder Arnold London, 3rd ed. / rev. by christian m.i.m. matthiessen. edn. (2004)
11. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
12. Křen, M., Cvrček, V., Čapka, T., Čermáková, A., Hnátková, M., Chlumská, L., Jelínek, T., Kováříková, D., Petkevic, V., Procházka, P., et al.: Syn2015: Representative corpus of contemporary written czech. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 2522–2528 (2016)
13. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)
14. Lehečka, J., Švec, J.: Comparison of czech transformers on text classification tasks. In: International Conference on Statistical Language and Speech Processing. pp. 27–37. Springer (2021)
15. Linkov, V., Smerk, P., Li, B., Smahel, D.: Personality perception in instant messenger communication in the czech republic and people's republic of china. Studia Psychologica **56**(4), 287 (2014)
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

17. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning–based text classification: A comprehensive review. ACM Computing Surveys (CSUR) **54**(3), 1–40 (2021)
18. Nick, E.A., Cole, D.A., Cho, S.J., Darcy K. Smith, T.G.C., Zelkowitz, R.: The online social support scale: Measure development and validation. Psychological assessment **30**(9), 1127–1143 (2018). https://doi.org/10.1037/pas0000558
19. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
20. Remschmidt, H., Nurcombe, B., Belfer, M., Sartorius, N., Okasha, A.: The Mental Health of Children and Adolescents: An area of global neglect. World Psychiatric Association, Wiley (2007), `https://books.google.cz/books?id=bENaj6hBuQUC`
21. Seznam.cz: Small-e-czech. `https://github.com/seznam/small-e-czech` (2021)
22. Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., Konopík, M.: Czert–czech bert-like model for language representation. arXiv preprint arXiv:2103.13031 (2021)
23. Smahel, D., Machackova, H., Mascheroni, G., Dedkova, L., Staksrud, E., Ólafsson, K., Livingstone, S., Hasebrink, U.: EU Kids Online 2020: survey results from 19 countries. EU Kids Online (2020)
24. Sotolář, O., Plhák, J., Šmahel, D.: Towards personal data anonymization for social messaging. In: International Conference on Text, Speech, and Dialogue. pp. 281–292. Springer (2021)
25. Straka, M., Náplava, J., Straková, J., Samuel, D.: Robeczech: Czech roberta, a monolingual contextualized language representation model. arXiv preprint arXiv:2105.11314 (2021)
26. Tuarob, S., Tucker, C.S., Salathe, M., Ram, N.: An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages. Journal of Biomedical Informatics **49**, 255–268 (2014). https://doi.org/https://doi.org/10.1016/j.jbi.2014.03.005, `https://www.sciencedirect.com/science/article/pii/S1532046414000628`
27. Underwood, M.K., Ehrenreich, S.E., More, D., Solis, J.S., Brinkley, D.Y.: The blackberry project: The hidden world of adolescents' text messaging and relations with internalizing symptoms. Journal of Research on Adolescence **25**(1), 101–117 (2015)
28. Vig, J.: Bertviz: A tool for visualizing multihead self-attention in the bert model. In: ICLR Workshop: Debugging Machine Learning Models (2019)
29. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
30. Yang, D., Flek, L.: Towards user-centric text-to-text generation: A survey. In: Ekštein, K., Pártl, F., Konopík, M. (eds.) Text, Speech, and Dialogue. pp. 3–22. Springer International Publishing, Cham (2021)

When Tesseract Brings Friends

Layout Analysis, Language Identification, and Super-Resolution in the Optical Character Recognition of Medieval Texts

Vít Novotný[1], Kristýna Seidlová[2], Tereza Vrabcová[1], and Aleš Horák[1]

[1] Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`{witiko,485431}@mail.muni.cz, hales@fi.muni.cz`
[2] Department of Auxiliary Historical Sciences and Archive Studies
Faculty of Arts, Masaryk University
Arna Nováka 1, 602 00 Brno, Czech Republic
`449852@mail.muni.cz`

Abstract. In our previous article, we surveyed optical character recognition algorithms for medieval texts. However, accurate recognition remains an open challenge. In this work, we develop eight preprocessing techniques and we show that they improve OCR accuracy on medieval texts. We also produce and publish an open dataset of 51,351 scanned images and OCR texts with 120 human annotations for layout analysis and OCR evaluation, and 122 human annotations for language identification.

Keywords: Optical character recognition · Layout analysis · Language identification · Image super-resolution · Medieval texts

1 Introduction

The aim of the AHISTO project is to make documents from the Hussite era (1419–1436) available to the general public through a web-hosted searchable database. Although scanned images of letterpress reprints from the 19th and 20th century are available, accurate optical character recognition (OCR) algorithms are required to extract searchable text from the scanned images.

In our previous article [15], we have shown that the Tesseract 4 OCR algorithm was the second fastest and the most accurate among five different OCR algorithms. In this article, we investigate the impact of six preprocessing techniques on the accuracy of Tesseract 4. Additionally, we compare Tesseract 4 with three other OCR algorithms on the language identification task. Furthermore, we publish an open dataset [16] of scanned images and OCR texts with human annotations for layout analysis, OCR evaluation, and language identification.

In Section 2, we describe the related work in OCR preprocessing. In Section 3, we describe our three preprocessing techniques and our two evaluation tasks. In Section 4, we discuss the results of our evaluation. In Section 5, we offer concluding remarks and ideas for future work in the OCR of medieval texts.

2 Related Work

Today's OCR algorithms use complex preprocessing pipelines that try to rid the scanned images of artefacts introduced by the printing process, the aging and

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 29–39, 2021. © Tribun EU 2021

degradation of the paper, and the scanning process. In our work, we introduce eight additional preprocessing techniques based on layout analysis, language detection, and image super-resolution. In this section, we discuss the related work in each of these three areas.

2.1 Layout Analysis

In OCR preprocessing, layout analysis is one of the first steps, where the page is divided into areas of text and non-text. Two main types of methods exist:

1. *Bottom-up* methods either classify small patches of the scanned images and cluster patches of the same class into larger areas [26,17,7,3,4] or analyze whitespace to detect boundaries between areas [1,19,2]. They can adapt to non-rectangular areas but they often miss the global structure of the page.
2. *Top-down* methods [27,11,12] slice the page recursively into horizontal and vertical strips. They can discover large rectangular areas such as headings, columns, and paragraphs, but may fail to segment non-rectangular areas.

Tesseract 4 uses a hybrid technique [23] that first uses bottom-up techniques to detect the smaller areas in the page and then uses top-down techniques to group the smaller areas and decide their reading order.

2.2 Language Identification

In order to improve their accuracy, OCR algorithms need to identify the language of the text, so that they can use dictionaries and language models to narrow down the number of possible readings of the text.

Tesseract optimizes character segmentation and language modeling[3] [22,10]. The hypothesis with the highest combined score determines the language of a word. Older versions of Tesseract used separate models for character segmentation and language modeling and only combined their scores. Tesseract 4 uses a LSTM model that jointly optimizes both criteria.[4]

2.3 Image Super-Resolution

Traditionally, OCR engines used simple rule-based methods to maximize the signal-to-noise ratio in scanned images. Recent results show that image super-resolution techniques based on deep neural networks such as SRCNN [5] and the more advanced SRGAN [9] can be used as a preprocessing technique that improves OCR accuracy [8,13,24,14,6,20]. For more information about image super-resolution techniques, see another article from these proceedings on page 11.

---

[3] https://tesseract-ocr.github.io/docs/das_tutorial2016/4CharSegmentation.pdf

[4] https://tesseract-ocr.github.io/docs/das_tutorial2016/7Building%20a%20Multi-Lingual%20OCR%20Engine.pdf

3 Methods

In this section, we describe the OCR algorithms that we use in our experiments. We also describe our preprocessing techniques and how we evaluate them. Our experimental code is available online.[5]

3.1 Optical Character Recognition

Besides Tesseract 4, we also use Tesseract 3, Tesseract 3 + 4, and Google Vision AI in our language identification experiments. We also use Google Vision AI in our image super-resolution experiments. For more information about the different OCR algorithms, see our previous article [15, Section 2].

3.2 Scanned Image Dataset

In our previous article, we developed a dataset [15, Section 3.1] of 65,348 scanned image pairs in both low resolution (150 DPI) and high resolution (400 DPI).

To make it easy for others to reproduce and build upon our work, we use a subset of 51,351 scanned images (79%) from public-domain books in our experiments and we publicly release our dataset [16].

3.3 Preprocessing

In this section, we describe our eight preprocessing techniques: two based on layout analysis, two based on layout identification techniques, and four based on image super-resolution.

Layout Analysis In our previous article, we showed that Google Vision AI [15, Section 4.2] is accurate but can fail to properly segment multi-column pages where Tesseract 4 does not.

We developed two layout analysis techniques based on *computational geometry* (see Algorithm 1) and *machine learning* (see Algorithm 2). We use our techniques to decide whether a page is single- or multi-column. Single-column pages are processed by Google Vision AI and multi-column pages by Tesseract 4.

---

[5] http://gitlab.fi.muni.cz/xnovot32/ahisto-ocr, file when-tesseract-brings-friends.ipynb

Algorithm 1: Layout analysis using computational geometry

```
Result: Whether the page contains a single column of text or multiple
Shoot seven horizontal rays in uniform vertical intervals over the page height;
Compute how many lines $l_i$ in OCR output each ray $i$ intersects;
if $median_{i \in \{2,3,\ldots,6\}} l_i \leq 1$ then
    The page contains a single column of text;
else
    The page contains multiple columns of text;
end
```

Algorithm 2: Layout analysis using machine learning

```
Result: Whether the page contains a single column of text or multiple
Collect the $x$-coordinates of left and right boundaries of all lines in OCR output;
Combine the collected left and right boundaries into a set $B$ of all boundaries;
Use sklearn.svm.OneClassSVM to remove outliers from $B$;
Find the best number $k \in \{0, 1, \ldots, \min(10, |B|)\}$ of $k$-means clusters of $B$ by
  maximizing the Silhouette score;
if $k \leq 2$ then
    The page contains a single column of text;
else
    The page contains multiple columns of text;
end
```

Algorithm 3: Language identification based on paragraph languages

```
Result: Probability distribution $\Pr(l)$ over the languages $l$ of the page
foreach candidate language $l$ do
    $\text{count}_l \leftarrow 0$;
end
foreach paragraph $p$ with language $l$ from the set of candidate languages do
    $\text{count}_l \leftarrow \text{count}_l +$ length of paragraph $p$ in characters;
end
foreach candidate language $l$ do
    $\Pr(l) \leftarrow \text{count}_l / \sum_{l'} \text{count}_{l'}$;
end
```

Language Identification In 2006, Panák [18, Section 4.4] showed that using two-pass processing, where we first identify languages and then use the OCR algorithm with the identified languages can improve OCR accuracy. We developed two techniques for identifying page language using the languages of *paragraphs* (see Algorithm 3) and *words* (see Algorithm 4) in the OCR output of Tesseract 4.

In the first pass, we identified page languages using Tesseract 4 with two different sets of candidate languages based on the most frequent languages in our dataset: *three* (Czech, German, and Latin) and *nine* (Czech, German, Latin, Polish, French, English, Russian, Italian, and Slovak) candidate languages.

In the second pass, we use Tesseract 4 with languages l that were *detected* ($\Pr(l) > 0\%$) and that satisfied $\Pr(l) \geq t$ for a number of different thresholds $t \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$. If none, then an empty OCR output is produced.

Image Super-Resolution The scanned images in the AHISTO project are often only available in the low resolution of 150 DPI. We use image super-resolution techniques to jointly upscale and reconstruct the images.

As our baseline preprocessing techniques, we use the original low-resolution and high-resolution images, and low-resolution images that were upscaled 2× using either bilinear interpolation or the Potrace vectorizer [21].

Algorithm 4: Language identification based on word languages

```
Result: Probability distribution $\Pr(l)$ over the languages $l$ of the page
foreach candidate language $l$ do
    $\text{count}_l \leftarrow 0$;
end
foreach paragraph $p$ with language $l$ from the set of candidate languages do
    $\text{count}_l \leftarrow \text{count}_l$ + length of paragraph $p$ in characters;
    foreach word $w \in p$ with language $l'$ from the set of candidate languages do
        $\text{count}_l \leftarrow \text{count}_l$ − length of word $w$ in characters;
        $\text{count}_{l'} \leftarrow \text{count}_{l'}$ + length of word $w$ in characters;
    end
end
foreach candidate language $l$ do
    $\Pr(l) \leftarrow \text{count}_l / \sum_{l'} \text{count}_{l'}$;
end
```

As our actual preprocessing techniques, we use low-resolution images upscaled either 2× using SRCNN or 4× using SRGAN. For SRCNN, we use two public SRCNN models[6] (further known as *Waifu2x*) that were pre-trained on drawn mange images with two different levels of noise removal: *low* (`noise0`) and *high* (`noise3`). For SRGAN, we use two models that we trained on the scanned images in our dataset and the born-digital PDF version of tome six of the book *Codex Diplomaticus et Epistolaris Regni Bohemiae* (further known as CDB VI) [25].

3.4 Evaluation

We evaluate our preprocessing techniques both intrinsically on the layout analysis and language detection tasks, and extrinsically on the OCR accuracy.

Layout Analysis For layout analysis, we report confusion matrices for the binary classification of pages as either single-column or multi-column. As our ground truth, we use 120 human-annotated pages that we publicly release in our dataset.

Language Identification For language identification, we report the percentage of pages (further known as *Accuracy@1*) where we correctly identified the primary language in the first pass. As our ground truth, we use 122 human-annotated pages[7] that we publicly release in our dataset.

Optical Character Recognition For OCR accuracy, we report the word error rate (further known as WER) [15, Section 3.2]. As our ground truth, we use 120 human-annotated pages that we publicly release in our dataset.

---

[6] https://github.com/nagadomi/waifu2x/tree/master/models/cunet/art

[7] https://gitlab.fi.muni.cz/nlp/ahisto-language-detection

4 Results

In this section, we report the results of our evaluation and we discuss the corpus of OCR texts that we created with our most successful preprocessing techniques,

4.1 Layout Analysis

Figure 1 shows that our simpler layout analysis technique that used computational geometry performed better on the intrinsic classification task and misclassified only two out of 120 ($1.\overline{6}\%$) pages. Our machine learning technique misclassified 31 out of 103 (30.1%) single-column pages as multi-column pages.

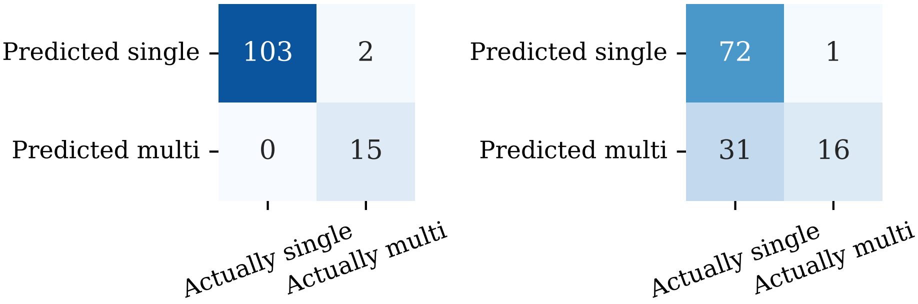


Fig. 1: Confusion matrices of computational geometry (left) and machine learning (right) layout analysis techniques

Figure 2 confirms our observation that although Google Vision AI performs generally worse than Tesseract 4, it performs significantly better on single-column pages and fails catastrophically on multi-column pages. By combining Google Vision AI and Tesseract 4 with our layout analysis technique usinf computational geometry, we receive significant improvements to the OCR accuracy.

4.2 Language Identification

Figure 3 shows that Google Vision AI performs significantly better than Tesseract on the intrinsic page language identification task. For Tesseract, using nine candidate languages with the *word* language identification technique consistently outperformed other configurations.

Figure 4 shows that using two-pass processing with nine candidate languages, the *paragraph* language identification technique that limits the number of detected languages, and the 0% threshold that only removes candidate languages that weren't at all detected can improve the OCR accuracy of Tesseract 4.

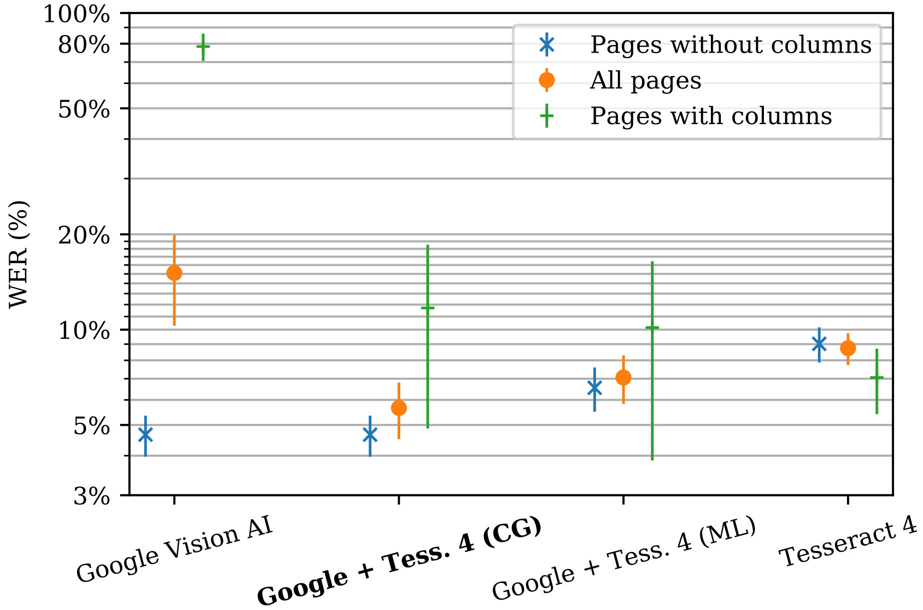


Fig. 2: OCR accuracies of Google Vision AI and Tesseract 4 alone and combined using two different layout analysis techniques (computational geometry and machine learning) on different subsets of pages. The best technique is **bold**.

4.3 Super-Resolution

Figure 5 shows that Google Vision AI does not particularly benefit from image super-resolution techniques. In contrast, Tesseract 4 always achieves better OCR accuracy with super-resolution techniques than with low-resolution images and outperforms even high-resolution images with the Waifu2x and SRGAN image super-resolution techniques. The pre-trained Waifu2x models outperform our SRGAN models, which may indicate a lack of training data.

4.4 Text Corpus

We combined our most successful preprocessing techniques: layout detection using computational geometry, two-pass processing with 0% threshold, nine candidate languages, and *paragraph* language identification technique, and the Waifu2x image super-resolution technique with high noise removal.

With the combined techniques, we achieved 5.42% WER compared to 8.74% with no preprocessing. Additionally, we also produced 51,351 OCR texts that we include in our dataset.

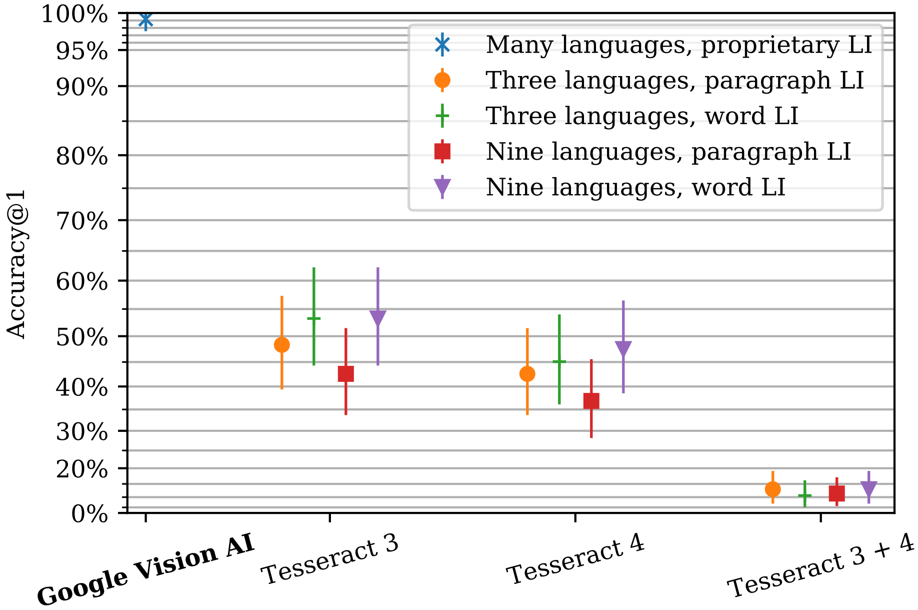


Fig. 3: Language identification accuracies of four different OCR engines using two different sets of candidate languages (three and nine) and two different language identification techniques (paragraph and word). The best OCR engine is **bold**.

5 Conclusion and Future Work

The OCR of scanned images for contemporary printed texts is widely considered a solved problem. However, the OCR of early printed books and reprints of medieval texts remains an open challenge. In our work, we developed eight preprocessing techniques in three different areas and we showed that they can improve the OCR accuracy on medieval texts. We also published an open dataset [16] of 51,351 scanned images and OCR texts with 120 annotations for layout analysis and OCR evaluation and 122 annotations for language identification.

In our work, we only used language identification preprocessing techniques based on language identification for individual pages. However, in printed collections of multilingual texts, OCR accuracy may be improved by processing smaller areas of the page separately. Additionally, we would produce an empty OCR output when no languages were detected or passed the confidence threshold, Just disabling the language models in Tesseract may give better results.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101 and by TAČR Éta, project number TL03000365. The first author's work was also funded

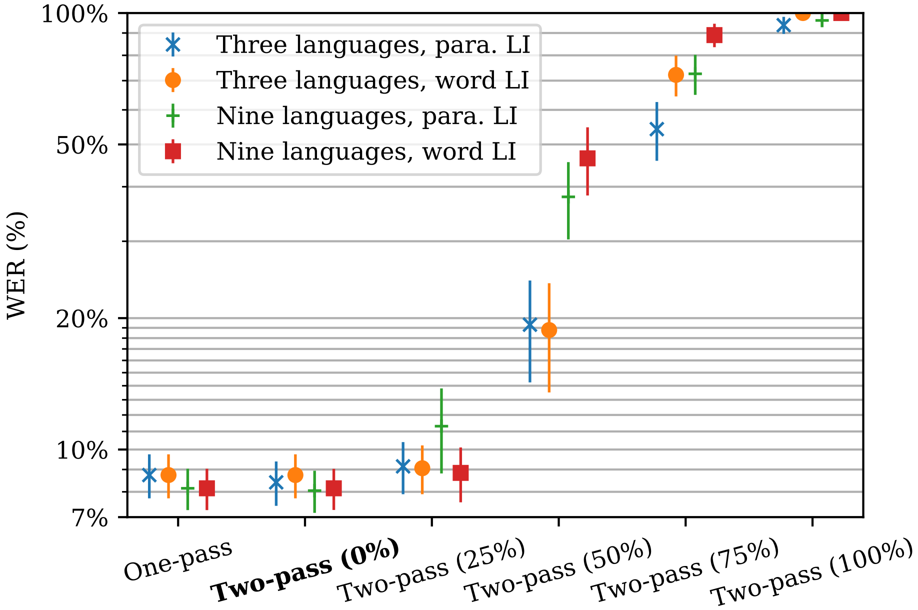


Fig. 4: OCR accuracies of Tesseract 4 using two different sets of candidate languages (three and nine) and two different page language identification techniques (paragraph and word). The best technique is **bold**.

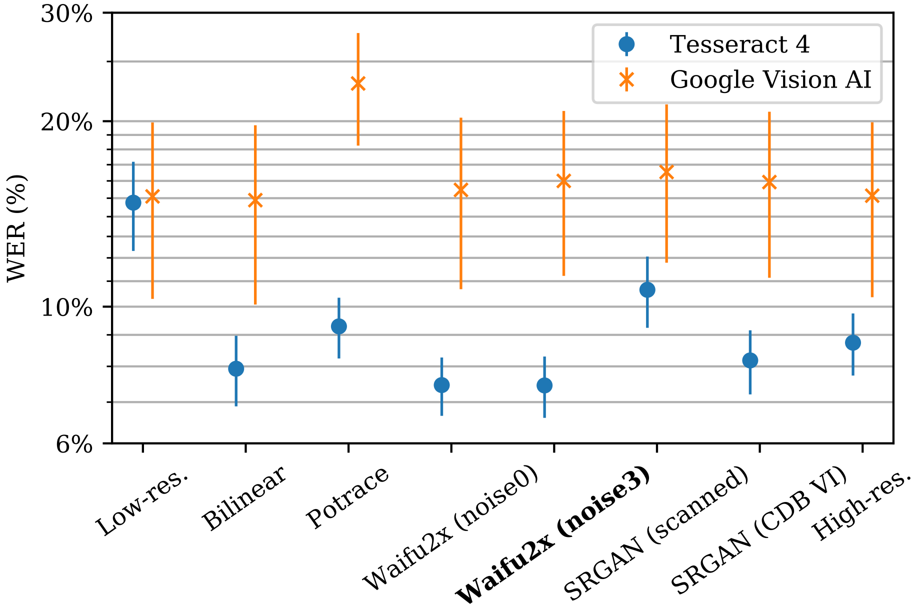


Fig. 5: OCR accuracies of Google Vision AI and Tesseract 4 using four different baselines and four different image super-resolution techniques. The best technique is **bold**.

by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project.

References

1. Baird, H.S., Jones, S.E., Fortune, S.J.: Image segmentation by shape-directed covers. In: ICPR. vol. 1, pp. 820–825. IEEE (1990)
2. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Int. workshop on document analysis systems. pp. 188–199. Springer (2002)
3. Chen, M., Ding, X., Wu, Y.: Unified HMM-based layout analysis framework and algorithm. Science in China, Series F: Information Sciences **46**(6), 401–408 (2003)
4. Chowdhury, S., Mandal, S., Das, A., Chanda, B.: Segmentation of text and graphics from document images. In: ICDAR. vol. 2, pp. 619–623. IEEE (2007)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 184–199 (2014)
6. Fu, Z., Kong, Y., Zheng, Y., Ye, H., Hu, W., Yang, J., He, L.: Cascaded detail-preserving networks for super-resolution of document images. In: ICDAR. pp. 240–245. IEEE Computer Society (2019)
7. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. Computer Vision and Image Understanding **70**(3), 370–382 (1998)
8. Lat, A., Jawahar, C.V.: Enhancing OCR accuracy with super resolution. In: ICPR. pp. 3162–3167. IEEE (2018)
9. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proc. of the IEEE conf. on computer vision and pattern recognition. pp. 4681–4690 (2017)
10. Lee, D.S., Smith, R.: Improving book OCR by adaptive language and image models. In: Int. Workshop on Document Analysis Systems. pp. 115–119. IEEE (2012)
11. Nagy, G., Seth, S.C.: Hierarchical representation of optically scanned documents. In: ICPR. pp. 347–349 (1984)
12. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. Computer **25**(7), 10–22 (1992)
13. Nakao, R., Iwana, B.K., Uchida, S.: Selective super-resolution for scene text images. In: ICDAR. pp. 401–406 (2019)
14. Nguyen, K.C., Nguyen, C.T., et al.: A character attention generative adversarial network for degraded historical document restoration. In: ICDAR. pp. 420–425 (2019)
15. Novotný, V.: When tesseract does it alone: Optical character recognition of medieval texts. In: Horák, A., Rychlý, P., Rambousek, A. (eds.) Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020. pp. 3–12. Tribun EU (2020)
16. Novotný, V., Seidlová, K., Vrabcová, T., Horák, A.: A human-annotated dataset of scanned images and OCR texts from medieval documents (2021), https://nlp.fi.muni.cz/projects/ahisto/ocr-dataset, [cited 2021-11-16]
17. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Transactions on pattern analysis and machine intelligence **15**(11), 1162–1173 (1993)
18. Panák, R.: Digitalizace matematických textů. Master's thesis, Faculty of Informatics, Masaryk University (2006), https://is.muni.cz/th/pspz5/
19. Pavlidis, T., Zhou, J.: Page segmentation and classification. CVGIP: Graphical models and image proc. **54**(6), 484–496 (1992)

20. Ray, A., Sharma, M., et al.: An end-to-end trainable framework for joint optimization of document enhancement and recognition. In: ICDAR. pp. 59–64 (2019)
21. Selinger, P.: Potrace: a polygon-based tracing algorithm (2003), http://potrace.sourceforge.net/potrace.pdf, [cited 2021-11-07]
22. Smith: An overview of the tesseract OCR engine. In: ICDAR. pp. 629–633. IEEE (2007)
23. Smith: Hybrid page layout analysis via tab-stop detection. In: ICDAR. pp. 241–245. IEEE (2009)
24. Su, X., Xu, H., Kang, Y., Hao, X., Gao, G., Zhang, Y.: Improving text image resolution using a deep generative adversarial network for optical character recognition. In: ICDAR. pp. 1193–1199 (2019)
25. Sviták, Z., Krmíčková, H., Krejčíková, J., Friedrich, G.: Codex diplomaticus et epistolaris Regni Bohemiae. Tomi VI. Academia (2006)
26. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Computer graphics and image proc. **20**(4), 375–390 (1982)
27. Wong, K.Y., Casey, R.G., Wahl, F.M.: Document analysis system. IBM journal of research and development **26**(6), 647–656 (1982)

Precomputed Word Embeddings for 15+ Languages

Ondřej Herman[1,2]

[1] Faculty of Informatics
Masaryk University
Botanická 68, 612 00 Brno
Czech Republic
`xherman1@fi.muni.cz`
[2] Lexical Computing s.r.o.
Botanická 68, 612 00 Brno
Czech Republic
`ondrej.herman@sketchengine.eu`

Abstract. We calculated word embedding models using fastText for multiple languages and corpora. The models are available for download and through a Web interface at `https://embeddings.sketchengine.eu/`.

Keywords: Word embeddings · Sketch Engine · Corpora

1 Word Embeddings

Word embeddings serve as an useful resource for many downstream natural language processing tasks. The embeddings map or embed the lexicon of a language onto a vector space, in which various operations can be carried out easily using the established machinery of linear algebra. The unbounded nature of the language can be problematic and word embeddings provide a way of compressing the words into a manageable dense space.

The position of a word in the vector space is given by the context the word appears in, or, as the distributional hypothesis postulates, *a word is characterized by the company it keeps* [2]. As similar words appear in similar contexts, their positions will also be close to each other in the embedding vector space. Because of this many useful semantical properties of words are preserved in the embedding vector space.

2 Models

The models were created using a modified version of the fastText [1] package with the ability to read corpora as indexed by the Manatee corpus manager, which is the core of the Sketch Engine [4]. This allows us to calculate models to have identical tokenization and format as the source corpora.

The models are calculated with a dimension of **100**, which is reasonable trade-off between size and performance for common applications. The minimum frequency for the lexicon elements has been chosen to be **5**, as for tokens

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 41–46, 2021. © Tribun EU 2021

with fewer appearances it is rarely possible to estimate quality word vectors. The **skip-gram** model has been chosen for the calculation. It is slightly more expensive to evaluate compared to the continuous-bag-of-words model, but the vector quality for rare words is improved. The negative-sampling parameter has been reduced to 3, as for large corpora this has negligible influence on the performance of the resulting model, while the training speed is greatly improved.

2.1 Source Corpora

Most of the models are based on the TenTen family of corpora [3]. These corpora have been built from texts obtained from the Web. The texts contained in the corpora are cleaned and deduplicated, and where available, the text is also available in lemmatized form and with part-of-speech annotations. The corpora can be accessed from the Sketch Engine[3].

For most of the corpora, multiple models are available. There is always a base model calculated from the **word** attribute, which represents the raw corpus text. A **lc** model is calculated from a lowercased variant of the corpus. A **lemma** model uses the corpus with every word converted to their base forms. A **lemma_lc** model is a lowercased variant of the **lc** model. A **lempos** model combines lemmata with a part-of-speech annotations appended. The Table 1 shows a selection of the models available with the respective lexicon sizes.

Table 1: Model Lexicon Sizes

Corpus	**lc**	**lemma**	**lemma_lc**	**lempos**	**word**
Arabic					2197469
Czech		2386157	2147712		3900455
Danish		1854619	1854541	1930823	2722811
German		6917255	7147030	6576701	6996045
Early English	799595	907219	776060	990898	962268
English	5929132	5941733	5268157	6143073	6658558
English (BNC2)		145773	130468	153041	200565
Spanish	3200355	2938116	2928086	3108981	3840913
Estonian	2915876	1906368			3307785
French	3581976	3971686	3304428	4300514	4335469
Italian	1325186	1363078	1134964	1508063	1624666
Korean					2949340
Portuguese	1872044	1700285	1700285	1783936	2264516
Russian	7494969	7770940	7205918	7858430	8340643
Slovenian	1143192	780745			1365370
Chinese					1636645

---

[3] `https://www.sketchengine.eu`

2.2 Data Format

The models are available for download in two different formats. Models with the `bin` extension are encoded in the native binary fastText format, while models with the `vec` extension use the textual Word2Vec format. We recommend the `bin` format, as it contains the subword n-gram information, is more compact and also faster to load.

2.3 Licensing

The models are available under the terms of the *Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License*[4]. This means that you can use the models for any non-commercial purposes and create derivative works based on the models, but you must give us credit and the derivative work needs to be available under the same terms.

3 Embedding Viewer

We also make the models accessible through a Web interface, which is hosted at `https://embeddings.sketchengine.eu/`. All the models which are available for download can also be examined through this interface.

The interface supports multiple types of queries. When a single word is entered, the words closest to it, according to cosine similarity, are retrieved and sorted by decreasing similarity.

When multiple words are entered, their word vectors are averaged and the result set consists of the words closest to the average value.

When a word in the query is prefixed with a minus ('-') character, the *inverse* of its word vector will be used, enabling to carry out arithmetic on the word vectors. For example, to obtain the result of *king* - *man* + *woman*, as formulated in [5], the user shall enter the query `king -man woman`. The result can be seen in the Figure 1.

3.1 API

In addition to the human-readable interface, the models can also be queried in an automated way and the result can be provided in machine-readable way. The supported formats are JSON and TSV.

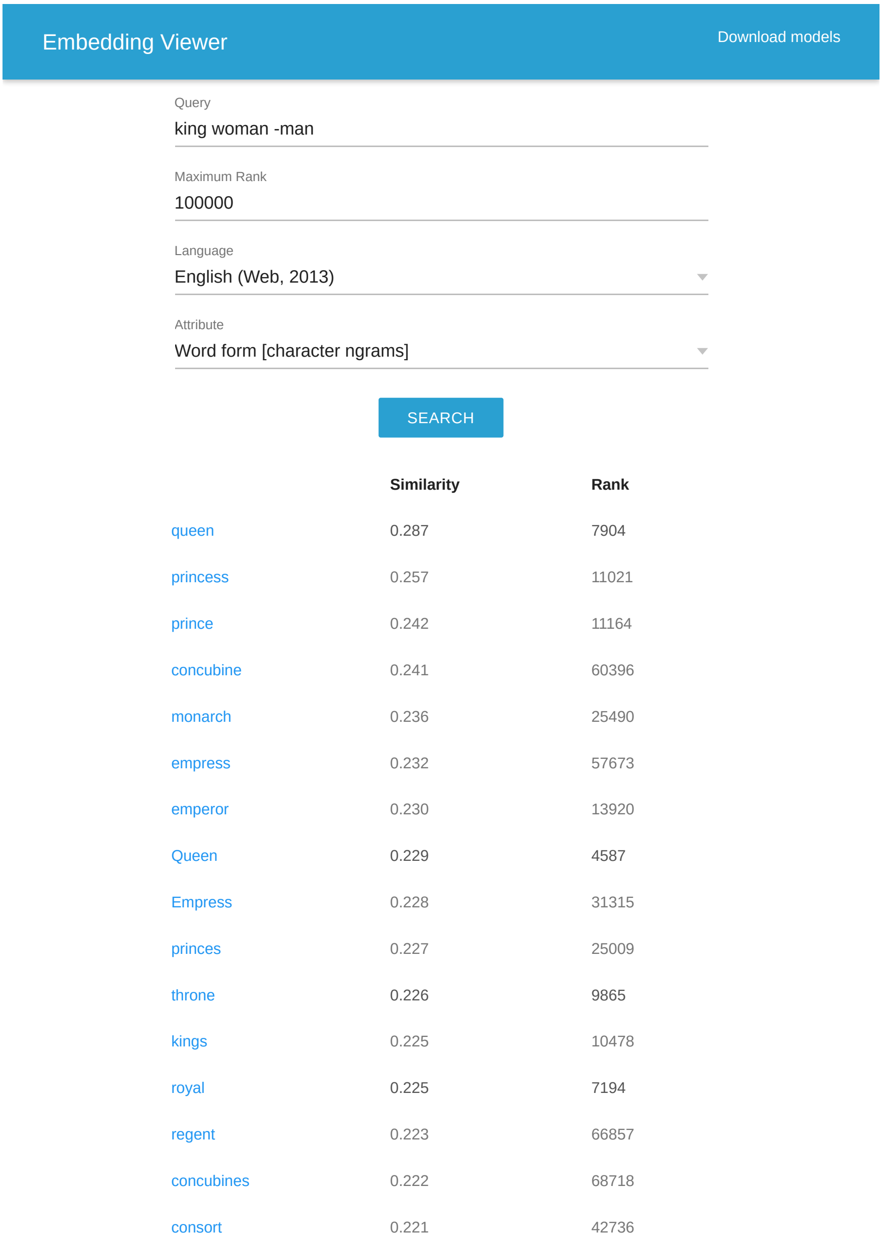
The endpoint at `https://embeddings.sketchengine.eu/` accepts the following parameters:

Providing at least one of the `q`, `pos` or `pos_vec` parameters is mandatory, other parameters are optional.

The parameters are identical to the ones generated by the HTML user interface, so a link copied from the browser provides a good starting point for further experiments. As an example, retrieving the top 5 most similar lemmata

---

[4] Avaliable at `https://creativecommons.org/licenses/by-nc-sa/4.0/`.



Query
king woman -man

Maximum Rank
100000

Language
English (Web, 2013)

Attribute
Word form [character ngrams]

	Similarity	Rank
queen	0.287	7904
princess	0.257	11021
prince	0.242	11164
concubine	0.241	60396
monarch	0.236	25490
empress	0.232	57673
emperor	0.230	13920
Queen	0.229	4587
Empress	0.228	31315
princes	0.227	25009
throne	0.226	9865
kings	0.225	10478
royal	0.225	7194
regent	0.223	66857
concubines	0.222	68718
consort	0.221	42736

Fig. 1: Embedding Viewer

Table 2: Embedding API Query Parameters

Parameter	**Description**
`q=QUERY`	a complete query formatted as described above
`pos=WORD`	a single query word, can be specified multiple times
`neg=WORD`	a single query word complement, can be specified multiple times
`pos_vec=VEC`	same as `pos`, but interpreted as a comma-separated vector
`neg_vec=VEC`	same as `neg`, but interpreted as a comma-separated vector
`n=N`	the amount of rows to be returned
`lim=N`	maximum rank of the result entries
`model=NAME`	name of the embedding model
`json`	format the result as JSON
`raw`	format the result as TSV (tab-separated columnar format)
`vec`	include the word vectors in the result

to the lemma *dog* according to the English (Web, 2013) model in tab-separated format can be carried out by the ‘curl‘ program[5].

```
$ curl 'https://embeddings.sketchengine.eu/?q=dog&lim=100000&n=5&
       model=English+%28Web%2C+2013%29%7CLemma&raw'

    puppy  0.8980982303619385 4139
    cat    0.8976492285728455 1678
    canine 0.8802799582481384 8694
    pup    0.8700659275054932 9166
    pet    0.8562509417533875 1622
```

Should you need lemmata similar to the lemma *cat* formatted as JSON, use the following query instead:

```
$ curl 'https://embeddings.sketchengine.eu/?q=cat&lim=100000&n=5&
       model=English+%28Web%2C+2013%29%7CLemma&json'

    {"w":[
        ["dog", 0.8976492881774902, 685],
        ["kitten", 0.8868610858917236, 8330],
        ["feline", 0.8669211864471436, 15259],
        ["pet", 0.8627837896347046, 1622],
        ["chinchilla", 0.8478652834892273, 51731]]
    }
```

The tab-separated format is easily usable for shell scripting and other similar ″free-form″ approaches, while JSON might be more appropriate for integration into more complex systems, in which the regular standardized form provides full control over the parsing details.

---

[5] Available from `https://curl.se/` for all common operating systems.

4 Future Work

The models which we have currently published cover only the most common languages. As we keep creating new corpora and extend existing ones, we will publish updated models in the future.

Of special interest might be models for other languages for which we have the data available. Eventually we plan to create word embedding models for every language present in the Sketch Engine. At the time of writing this article, this amounts to over 100 languages.

5 Conclusion

We calculated word embedding models using fastText for multiple languages and corpora. The models are available for download and through a Web interface at `https://embeddings.sketchengine.eu/`.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
2. Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954)
3. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The tenten corpus family. In: 7th International Corpus Linguistics Conference CL. pp. 125–127 (2013)
4. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The sketch engine: ten years on. Lexicography **1**(1), 7–36 (2014)
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)

Part II

Semantics and Language Modelling

Using FCA and Concept Explications for Finding an Appropriate Concept

Marek Menšík[1], Adam Albert[1], and Tomáš Michalovský[2]

[1] VSB - Technical University of Ostrava, Czech Republic
mensikm@gmail.com, adam.albert@vsb.cz,
[2] Silesian University in Opava, Czech Republic
mic0129@slu.cz

Abstract. This paper introduces the method of discovering a plausible atomic concept that corresponds to the generated molecular concept explication and known attributes' values and properties of objects falling under the concept. First, we summarize the process of concept explication via the symbolic method of supervised machine learning from formalized natural language sentences. To obtain particular concept explications, we exploit heuristic procedures that operate on the symbolic representation of current hypothesis and example to obtain particular concept explications. These explications serve as descriptions of the sought atomic concept accordingly to the given text sources. Afterwards, the method of searching for the appropriate concept based on attributes' values is outlined. Thus, user can seek a specific concept, which can be vague or inaccurate, among the so-extracted explications. We focus on a situation in which the user knows basic properties or attributes' values and searches for a suitable atomic concept that is described by these properties or attributes' values. To explain the process, we summarize the creation of explications and the method of Formal Concept Analysis (FCA) as a theoretical background. As a result, we present to the user an appropriate atomic concept. The whole method is demonstrated by a few examples.

Keywords: FCA · NLP · Explications · Formal concept

1 Introduction

The paper is follow up to our current natural language processing research. In [1], we exploited the supervised machine learning for creation of hypothesis that classifies objects. In [2], we modified the algorithm of machine learning for concept refinement in the form of explications obtained from texts in natural language. In [3], the method of seeking appropriate text sources was presented.

In this paper, we deal with the method of recommending an appropriate concept by a given specific set of properties or values of attributes of objects that are falling under the concept. To this date, we have dealt with creating of explications and with the recommendation of a relevant text source based on a chosen explication. In this paper, we decided to reverse this process and by

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 49–60, 2021. © Tribun EU 2021

exploiting FCA we seek a concept that corresponds to the given set. To introduce the reader to this problem, the explication is explained in the beginning.

The explication [4] is a process of refining of an inaccurate or vague expression into an adequately accurate one. For a sake of simplicity, we will refer to refinement as the concept explication.

In prior papers, we have focused on creating of explications of concepts using symbolic methods of supervised machine learning, that utilizes induction heuristics. These functions manipulates with a symbolic representation of explication in process of learning. As for symbolic representation we chose the strong expressiveness of Transparent intensional logic (TIL) and it's computational variant the TIL-script language. TIL and TIL-script are thoroughly described in publications such as [5], [6]. For this reason we will not explain them but we will highlight their features we exploited.

This paper is structured as follows. In chapters 2, the process of creating of explications is described. Chapter 3 summarizes the theory of FCA needed for understanding the *aspirant ordering* used for ordering of concepts by relevancy to the user. In chapter 4 we present the whole process of finding *appropriate concepts* on an example and in last chapter 5 concludes our research.

2 Supervised Machine Learning

Supervised machine learning is a method in which an agent is being trained by classified training examples provided by the supervisor. Examples are described by attributes divided into two groups, namely input and output attributes. There is a functional dependency f between values of those two groups. For example, conditions for receiving a loan by bank can be described by input attributes *employment*, *salary*, *age*, *indebtedness* and *health condition* of an applicant. The risk of providing a loan to the applicant is the output attribute. The goal of the supervised machine learning is that the agent creates his own functional dependency h by observing values of input and output attributes. Agent's functional dependency h, called a *hypothesis*, should approximate the original unknown function f.

Correctness of the learned hypothesis is verified by special set of examples called test examples. The agent knows only the values of input attributes of the test examples. If the hypothesis predicts the same values of output attributes as the original dependency f, the hypothesis is correct. More about supervised machine learning can be found in [7], [8], [9].

2.1 Algorithm framework

As one of the symbolic methods of supervised machine learning, our algorithm can be described by its general framework [8]. This framework consists of four parts: objectives, training data, data representation, and a module that operates on the symbolic representation.

Our adjusted algorithm does not produce hypothesis which would correctly classify unknown examples; it rather builds an explication of an atomic concept C. In TIL, the atomic concept is a Trivialisation of an object X, in symbols, $'X$.[3] The objective of our algorithm is to create the explication an explication in the form of a closed molecular construction producing an object $'Y$ as close to object X as possible.

The natural language sentences mentioning the atomic concept C play the role of training data. Sentences satisfying this condition are formalized into the language of TIL constructions, which serve as a data representation for our algorithm.

The module for manipulation with the symbolic representation contains heuristic functions. For our purpose, we have chosen functions from Patrick Winston's algorithm [10] adjusted for natural language processing. They are divided into two categories of functions.

Functions from the *Generalization* category replace one or more constituents of the hypothesis by a more general one. New or adjusted constituent is either created based on agent's internal ontology or it is created as a disjunction of new and existing constituents. In case of numerical values in the existing constituent and example, generalization can create an interval spanning both numerical values or it can alter existing numerical interval to cover the new value in example. For example, if we have a piece of information in the hypothesis that lions can live up to 10 years on average in the wild and in the example, we have another piece of information that lions can live up to 14 years on average in the wild, generalization will adjust the information in the hypothesis that lions can live up from 10 to 14 years on average in the wild. Thus, our hypothesis becomes more general.

Specialization is triggered by negative examples. In this case, new constituent is inserted into the molecular hypothesis. The constituent doesn't belong into the essence of an explicated object but it helps to distinguish the hypothesis from other similar explications. For instance, the explication of lioness can be specialized with a constituent meaning that lioness does not have a mane. With this information, we can differentiate the explication of lioness from for example an explication of lion.

The original Winston's algorithm [10] deals with examples that cover all the attributes of a learned object. It was not suitable for processing natural-language texts. Sentences that mention explicated object usually do not contain all requisites or typical properties of the object. Since we need to insert new constituents into the explication, we introduced in [2] a new algorithm method called *Refinement*, which contains a single heuristic function for adding a new requisites or typical properties into the hypothesis. More about heuristic functions contained in the generalization, specialization and refinement and about the process of creating explications can be found in [3].

---

[3] Trivialization $'X$ can be found in other papers written as ^0X.

2.2 Example of Generating an Explication

The symbolic methods of supervised machine learning that we discussed in [1] use heuristic functions which manipulate with a symbolic representation of the hypothesis to obtain the correct one. The language of TIL constructions was chosen for the symbolic representation of hypothesis and examples. This method was then adjusted in [2] for the purpose of the explication of an atomic concept by extracting sentences in natural language texts mentioning the atomic concept as positive and negative examples. Input attributes were in the form of molecular concepts explicating the learnt concept. Output attribute was the atomic concept to be learned. For example, to explicate a atomic concept *lioness*, i.e. Trivialization *'Lioness* of the property of being a lioness of type $(o\iota)_{\tau\omega}$, we can use sentences in natural language which explicate the property. For example, the positive example *"Lioness is a mammal which is an apex predator"*. The property of *being a mammal and apex predator* is formalized in TIL as the following construction.

$$\lambda w \lambda t\, \lambda x\, [[{}'Mammal_{wt}\; x] \wedge [[{}'Apex\; {}'Predator]_{wt}\; x]]$$

Types: *Apex*$/((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$: property modifier;
Predator, Mammal$/(o\iota)_{\tau\omega}$: properties of individuals; $w \rightarrow \omega; t \rightarrow \tau; x \rightarrow \iota$: variables ranging over possible worlds, times and individuals, respectively. TIL and its utilization in the process of explication is described in detail in [2], [3]. Reader can find more about TIL itself in [5], [12].

Training data for our method are natural language sentences. Only sentences mentioning the atomic concept are extracted and formalized into TIL constructions. Agent's hypothesis is refined or generalized by exploiting positive examples. By refinement, we insert new constituents into the hypothesis. With generalization, we adjust current constituents to prevent over specialization of the explication. By negative examples, we specialize the hypothesis to differentiate it from other similar concepts. For example, we can refine the above mentioned explication mentioned above with a positive example in the form of the sentence *"The lioness has a fur."*. The property of *having a fur* formalized in TIL construction as

$$\lambda w \lambda t\, \lambda x\, [{}'Has\text{-}fur_{wt}\; x]$$

Types: *Has-fur*$/(o\iota)_{\tau\omega}$: property of individuals.

This positive example triggers a heuristic function that enriches the hypothesis with a new constituent in conjunctive way.

$$\lambda w \lambda t\, \lambda x\, [[{}'Mammal_{wt}\; x] \wedge [[{}'Apex\; {}'Predator]_{wt}\; x] \wedge [{}'Has\text{-}fur_{wt}\; x]]$$

As mentioned above, by generalization, we can avoid having the explication too specific. For example, the explication contains an information that lioness lives in Africa.

$$\lambda w \lambda t\, \lambda x[[{}'Mammal_{wt}\ x] \wedge [[{}'Apex\ {}'Predator]_{wt}\ x]$$
$$\wedge\ [{}'Has\text{-}fur_{wt}\ x] \wedge [{}'Lives\text{-}in_{wt}\ x\ \lambda w \lambda t\, \lambda y[{}'Africa_{wt}\ y]]]$$

Types: $Lives\text{-}in/(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$; $Africa/(o\iota)_{\tau\omega}$

The explication can be generalized by a positive example *"Lioness lives in India."*. Generalization will adjust existing constituent in disjunctive way, thus making the explication more general.

$$\lambda w \lambda t\, \lambda x[[{}'Mammal_{wt}\ x] \wedge [[{}'Apex\ {}'Predator]_{wt}\ x]$$
$$\wedge\ [{}'Has\text{-}fur_{wt}\ x] \wedge [[{}'Lives\text{-}in_{wt}\ x\ \lambda w \lambda t\, \lambda y[{}'Africa_{wt}\ y]]$$
$$\vee\ [{}'Lives\text{-}in_{wt}\ x\ \lambda w \lambda t\, \lambda y[{}'India_{wt}\ y]]]]$$

Types: $India/(o\iota)_{\tau\omega}$

3 FCA and Aspirant Ordering

As stated above, the user selects the set of properties and attributes' that should characterise the sought concept. To this end, we exploit the FCA theory that is described in this chapter. The FCA is utilized to obtain all formal concepts and create conceptual lattice over explications.[4] The lattice provides overview of explication ordering. Base on the set of formal concepts we find all *'concept aspirants'*. Concept Aspirants (CA) is the set union of all concepts' intents of which the selected set of properties is an intents' subset. Next the set is ordered and the maximal element of the set is presented to the user as the *most appropriate* one.

As we mentioned in [6]. Formal Conceptual Analysis (FCA) was introduced in 1980s by the group lead by Rudolf Wille and became a popular technique within the information retrieval field.[5] FCA has been applied in many disciplines such as software engineering, machine learning, knowledge discovery and ontology construction. Informally, FCA studies how objects can be hierarchically grouped together with their mutual common attributes.

The following part deals with formal definitions and examples describing the process of selecting the most appropriate concept.

Definition 1. *Let* (G,M,I) *be a formal context, then* $\beta(G,M,I) = \{(O,A)|O \subseteq G, A \subseteq M, A^{\downarrow} = O, O^{\uparrow} = A\}$ *is a set of all formal concepts of context* (G,M,I) *where* $I \subseteq G \times M, O^{\uparrow} = \{a|\forall o \in O, (o,a) \in I\}, A^{\downarrow} = \{o|\forall a \in A, (o,a) \in I\}$. $A^{\downarrow}$ *is called* ***extent*** *of formal concept* (O,A) *and* $O^{\uparrow}$ *is called* ***intent*** *of formal concept* (O,A).

Definition 2. ***Concept aspirants*** *of the set of attributes* $\boldsymbol{a}$ *in* $\beta(G,M,I)$ *is a set* $CA(a) = \bigcup_{i=1}^{n} O_i^a$, *where* O^a *is extent of a concept* $(O,A) \neq (G,B), a \subseteq A, B \subseteq M$. *Namely,* ***concept aspirants*** *of the set of attributes* $\boldsymbol{a}$ *is a union of all formal concept extents where* $\boldsymbol{a}$ *is a subset of a particular formal concepts' intents.*

---

[4] In this paper we do not visualise the concept lattice as a graph structure.

[5] More in [13].

Definition 3. *Let $CA(a)$ be a set of concept aspirants of a set of attributes* ***a****, let $\delta(a)$ be a set of concepts (O, A) where $a \subseteq A$, i.e.: $\delta(a) = \{(O^a, (O^a)^\uparrow) | (O^a, (O^a)^\uparrow) \neq (G, B), B \subseteq M, (O^a, (O^a)^\uparrow) \in \beta(G, M, I)\}$. Then $\boldsymbol{x} \sqsubseteq \boldsymbol{y}$ is in relation of* ***aspirant ordering*** *iff $max(|(O^y)^\uparrow|) \leq max(|(O^x)^\uparrow|), x, y, \in CA(a), (O^x, (O^x)^\uparrow), (O^y, (O^y)^\uparrow) \in \delta(a)$.*

Definition 4. *Let $(CA(a), \sqsubseteq)$ be an ordered set according to the definition 3, then the maximal elements are the* ***most appropriate concepts****.*

Example: Let us have a formal context described by the following Table 1 and assume that the user seeks a concept which is described by the set of attributes $a = \{a_2\}$.

Table 1: Formal context

	a_0	a_1	a_2	a_3
o_0	1	1	0	0
o_1	0	1	1	0
o_2	0	1	1	1

The set of all *formal concepts*

$$(G, M, I) = \{C_0, C_1, C_2, C_3, C_4\},$$

where

$$\begin{array}{ll} C_0 = (\{o_0, o_1, o_2\}, \{a_1\}) & C_1 = (\{o_0\}, \{a_0, a_1\}) \\ C_2 = (\{o_1, o_2\}, \{a_1, a_2\}) & C_3 = (\{o_2\}, \{a_1, a_2, a_3\}) \\ C_4 = (\emptyset, \{a_0, a_1, a_2, a_3\}) & \end{array}$$

Find the set of *concept aspirants* for attributes *a*

$$a = \{a_2\}$$

1. Find set $\delta(a)$:
   $\delta(a) = \{(\{o_1, o_2\}, \{a_1, a_2\}), (\{o_2\}, \{a_1, a_2, a_3\}),$
   $(\emptyset, \{a_0, a_1, a_2, a_3\})\}$
2. Create the union of all extents found in step 1 : $CA(\{a_2\}) = \{o_1, o_2\}$
3. For all $x \in CA(\{a_2\})$ calculate max of $|(O^x)^\uparrow|$, where $((O^x), (O^x)^\uparrow) \in \delta(a) \rightarrow max(|\{o_1, o_2\}^\uparrow|) = 2, max(\{|\{o_1, o_2\}^\uparrow|, |o_2{}^\uparrow|\}) = 3$
4. Order $CA(\{a_2\})$ by definition 3 $\rightarrow o_2 \sqsubseteq o_1$

Table 2: Aspirants' ordering

Exp.	Intent	DF
o_1	$\{a_1, a_2\}$	$\{a_1\}$
o_2	$\{a_1.a_2, a_3\}$	$\{a_1, a_3\}$

In our table the column DF represents the difference from the selected set of attributes $\{a_2\}$. The maximum entities according to the orderings are representatives of the most general formal concepts. The most appropriate concept is o_1.

4 Data-set of our Case Study

In this chapter, we present specific explications obtained from several text sources by the algorithm described in chapter 2. Presented explications deal with several concepts of feline predators. These explications are particular samples of all possible explications we can obtain from textual data, because we can obtain several explications of the same concept from different sources. For example, one explication can describe a lioness from an anatomical perspective, another resource may describe the environment in which lioness lives, and still another document describes its behaviour.

The advantage of using the expressive apparatus of TIL is obvious here, since the analyses of sentences that mention the explicated concept are so fine-grained that they are easy to read and understand. Thus, users can easily analyse the differences between particular molecular concepts explicating the target concept. For instance if there are some inconsistencies between the so-obtained explications, the user may exclude those that are not acceptable for him/her. Thanks to this approach, the selection is not based only on syntactic features like the occurrence of a given term, but also on semantic features provided by the fine-grained analysis.

Explications are built up by applying the relation *Typ-p* and the relation *Req* of type $(o(o\iota)_{\tau\omega})(o\iota)_{\tau\omega})$. *Typ-p* is the relation between properties P and Q such that *typically*, if an individual happens to be a Q then most probably it has the property P. For example, the property of *living in Africa* is a typical property of the property of *being a lioness*. On the other hand, *Req*(*uisite*) is a *necessary* relation between properties. Necessarily, if an individual happens to be a lioness, then it must be a mammal as well.

In our example we had at our disposal six explications of atomic concepts, namely explications describing the concepts of 'House cat', 'Jungle cat', 'Sand cat', 'Lynx', 'Lion' and 'Tiger'. All these explications were generated from various sentences formalized into the TIL constructions.

Selected sentences describing the concept 'House Cat':

The house cat is a mammal. The house cat has a fur. The house cat is domesticated. The average height of the house cat is 30cm.

House_Cat =

$$\begin{aligned}&[['\mathit{Req}\ '\mathit{Mammal}\ ['\mathit{House}\ '\mathit{Cat}]] \wedge ['\mathit{Req}\ '\mathit{Has\text{-}fur}\ ['\mathit{House}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Req}\ '\mathit{Domesticated}\ ['\mathit{House}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=\ [['\mathit{Avg}\ '\mathit{Height}]_{wt}\ x]\ '30]\ ['\mathit{House}\ '\mathit{Cat}]]]\end{aligned}$$

The jungle cat is a mammal. The jungle cat has a fur. The average body length of the jungle cat is from 55 to 112 cm. The average height of the jungle cat is 36,5 cm. The fur color of the jungle cat is brown.

Jungle_Cat =

$$\begin{aligned}&[['\mathit{Req}\ '\mathit{Mammal}\ ['\mathit{Jungle}\ '\mathit{Cat}]] \wedge ['\mathit{Req}\ '\mathit{Has\text{-}fur}\ ['\mathit{Jungle}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x[['\!\leq\ ['\mathit{Bd\text{-}lgth}_{wt}\ x]\ '112]\\ &\wedge\ ['\!\geq\ ['\mathit{Bd\text{-}lght}_{wt}\ x]\ '55]]\ ['\mathit{Jungle}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=\ [['\mathit{Avg}\ '\mathit{Height}]_{wt}\ x]\ '36.5]\ ['\mathit{Jungle}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=_p\ ['\mathit{Fur\text{-}color}_{wt}\ x]\ '\mathit{Brown}]\ ['\mathit{Jungle}\ '\mathit{Cat}]]]\end{aligned}$$

The sand cat is a mammal. The sand cat has a fur. The average body length of the sand cat is from 39 to 57 cm. The average height of the sand cat is 27 cm. The fur color of the sand cat is brown.

Sand_Cat =

$$\begin{aligned}&[['\mathit{Req}\ '\mathit{Mammal}\ ['\mathit{Sand}\ '\mathit{Cat}]] \wedge ['\mathit{Req}\ '\mathit{Has\text{-}fur}\ ['\mathit{Sand}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x[['\!\leq\ ['\mathit{Bd\text{-}lgth}_{wt}\ x]\ '57]\\ &\wedge\ ['\!\geq\ ['\mathit{Bd\text{-}lght}_{wt}\ x]\ '39]]\ ['\mathit{Sand}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=\ [['\mathit{Avg}\ '\mathit{Height}]_{wt}\ x]\ '27]\ ['\mathit{Sand}\ '\mathit{Cat}]]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=_p\ ['\mathit{Fur\text{-}color}_{wt}\ x]\ '\mathit{Brown}]\ ['\mathit{Sand}\ '\mathit{Cat}]]]\end{aligned}$$

The lynx is a mammal. The lynx has a fur. The body length of the lynx is less than 148 cm. The average height of the lynx is 75 cm. The lynx is the biggest European feline predator.

Lynx =

$$\begin{aligned}&[['\mathit{Req}\ '\mathit{Mammal}\ '\mathit{Lynx}] \wedge ['\mathit{Req}\ '\mathit{Has\text{-}fur}\ '\mathit{Lynx}]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!\leq\ [['\mathit{Avg}\ '\mathit{Bd\text{-}lgth}]_{wt}\ x]\ '148]\ '\mathit{Lynx}]\\ &\wedge ['\mathit{Typ\text{-}p}\ \lambda w \lambda t\ \lambda x['\!=\ [['\mathit{Avg}\ '\mathit{Height}]_{wt}\ x]\ '75]\ '\mathit{Lynx}]\\ &\wedge ['\mathit{Typ\text{-}p}\ \ ['\mathit{Biggest}\ \ ['\mathit{EU}\ \ ['\mathit{Feline}\ '\mathit{Predator}]]]\ \ '\mathit{Lynx}]]\end{aligned}$$

The lion is a mammal. The lion has a fur. The lion has a mane. The body length of the lion is from 170 to 250 cm.

Lion =

$$\begin{aligned}&[[{}'Req\ {}'Mammal\ {}'Lion] \wedge [{}'Req\ {}'Has\text{-}fur\ {}'Lion]\\ &\wedge [{}'Req\ {}'Pantherinae\ {}'Lion]\\ &\wedge [{}'Typ\text{-}p\ {}'Has\text{-}mane\ \ {}'Lion] \wedge [{}'Req\ [{}'Significant\ {}'Sex\text{-}Dimorph]\ {}'Lion]\\ &\wedge [{}'Typ\text{-}p\ \lambda w \lambda t\ \lambda x[[{}'\leq [{}'Bd\text{-}lgth_{wt}\ x]\ {}'250]\\ &\wedge\ [{}'\geq [{}'Bd\text{-}lght_{wt}\ x]\ {}'170]]\ {}'Lion]]\end{aligned}$$

The tiger is a mammal. The tiger has a fur. The tiger is an apex predator. The average height of the tiger is 117 cm.

Tiger =

$$\begin{aligned}&[[{}'Req\ {}'Mammal\ {}'Tiger] \wedge [{}'Req\ {}'Has\text{-}fur\ {}'Tiger]\\ &\wedge [{}'Req\ {}'Pantherinae\ {}'Tiger] \wedge [{}'Typ\text{-}p\ [{}'Apex\ {}'Predator]\ {}'Tiger]\\ &\wedge [{}'Typ\text{-}p\ \lambda w \lambda t\ \lambda x[{}'= [[{}'Avg\ {}'Height]_{wt}\ x]\ {}'117]\ {}'Tiger]]\end{aligned}$$

Types:

Req, *Typ-p* $/(o(o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$, *Bd-Lgth*, *Height* $/(\tau\iota)_{\tau\omega}$: attributes
Avg $/((\tau\iota)_{\tau\omega}(\tau\iota)_{\tau\omega})$: attribute modifier
Mammal, Cat, Has-Fur, Domesticated, Fur-color, Brown, Lynx, Predator, Lion, Pantherinae, Has-mane, Sex-Dimorph, Tiger $/(o\iota)_{\tau\omega}$: properties
$=_p\ /(o(o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$
$=, \leq, \geq\ /(o\tau\tau)$
Jungle, House, Sand, Feline, EU, Biggest, Apex, Significant $/((o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$: property modifiers
$\wedge/(ooo)$
$x \rightarrow \iota$

Seeking the appropriate concept:

At this point we demonstrate the method of dealing with the explications as described in the previous chapter. Having the above introduced explications obtained from natural-language sentences, all the constituents are extracted and arranged into the incidence matrix. Due to lack of space, incidence matrix is represented by transactions in table 3.[6]

Remark: Each object O in table 3 represents one explication of a particular natural language concept. The set of all subconstructions (attributes in table 3) represents *intent* of a particular formal concept. There exists a formal concept $(\{c\}, \{c\}^{\uparrow})$ for each explicated atomic concept c.

Using FCA, all formal concepts were obtained. List of 10 obtained formal concepts is presented in table 4. Due to lack of space in table 4 symbol O represents the set of all objects, i.e. $O = \{JC, SC, HC, Ly, Li, Ti\}$.

All mentioned attributes $A = \{a_1, \dots, a_{18}\}$ in table 3 represent the following properties in table 5.

---

[6] More details in [11]

Table 3: Explications and attributes

Explication (O)	Attributes (A)
Jungle cat (JC)	$\{a_1, a_2, a_3, a_4, a_5\}$
Sand cat (SC)	$\{a_1, a_2, a_4, a_6, a_7\}$
House cat (HC)	$\{a_1, a_2, a_8, a_9\}$
Lynx (Ly)	$\{a_1, a_2, a_{10}, a_{11}, a_{12}\}$
Lion (Li)	$\{a_1, a_2, a_{13}, a_{14}, a_{17}, a_{18}\}$
Tiger (Ti)	$\{a_1, a_2, a_{15}, a_{16}, a_{17}\}$

Table 4: Table of all formal concepts

C	Extent	Intent
C_1	O	$\{a_1, a_2\}$
C_2	$\{JC, SC\}$	$\{a_1, a_2, a_4\}$
C_3	$\{Li, Ti\}$	$\{a_1, a_2, a_{17}\}$
C_4	$\{HC\}$	$\{a_1, a_2, a_8, a_9\}$
C_5	$\{JC\}$	$\{a_1, a_2, a_3, a_4, a_5\}$
C_6	$\{SC\}$	$\{a_1, a_2, a_4, a_6, a_7\}$
C_7	$\{Ly\}$	$\{a_1, a_2, a_{10}, a_{11}, a_{12}\}$
C_8	$\{Ti\}$	$\{a_1, a_2, a_{15}, a_{16}, a_{17}\}$
C_9	$\{Li\}$	$\{a_1, a_2, a_{13}, a_{14}, a_{17}, a_{18}\}$
C_{10}	$\emptyset$	A

Assume that the user chooses the attribute a_{17} representing the property of being a *'Pantherinae'* and wants to know, which concept is represented by the chosen attribute most appropriately.

Concept aspirants are found according to definition 2.

$$CA(\{a_{17}\}) = \{Li, Ti\}$$

Afterward, the set $CA(\{a_{17}\})$ is ordered according to definition 3. The final ordering is as follows:

$$Li \sqsubseteq Ti$$

According to definition 4 the entity *Ti* is is a maximal one, and thus the concept of *'being a Tiger'* is presented to the user as the most appropriate one.

5 Conclusion

In this paper, we have described the method of finding an appropriate concept based on properties and attributes' values known by user. The method is based on data mining method of Formal Conceptual Analysis over explications created by the supervised machine learning algorithm. In the beginning, descriptions of concepts, called explications, are created using formalized natural language

Table 5: The list of all properties

a_1	$'\mathit{Mammal}$
a_2	$'\mathit{Has\text{-}fur}$
a_3	$\lambda w\lambda t\, \lambda x[['\leq ['\mathit{Bd\text{-}lgth}_{wt}\, x]\, '112] \wedge ['\geq ['\mathit{Bd\text{-}lght}_{wt}\, x]\, '55]]$
a_4	$\lambda w\lambda t\, \lambda x['=_p ['\mathit{Fur\text{-}color}_{wt}\, x]\, '\mathit{Brown}]$
a_5	$\lambda w\lambda t\, \lambda x['= [['\mathit{Avg}\ '\mathit{Height}]_{wt}\, x]\, '36.5]$
a_6	$\lambda w\lambda t\, \lambda x[['\leq ['\mathit{Bd\text{-}lgth}_{wt}\, x]\, '57] \wedge ['\geq ['\mathit{Bd\text{-}lght}_{wt}\, x]\, '39]]$
a_7	$\lambda w\lambda t\, \lambda x['= [['\mathit{Avg}\ '\mathit{Height}]_{wt}\, x]\, '27]$
a_8	$'\mathit{Domesticated}$
a_9	$\lambda w\lambda t\, \lambda x['= [['\mathit{Avg}\ '\mathit{Height}]_{wt}\, x]\, '30]$
a_{10}	$\lambda w\lambda t\, \lambda x['\leq [['\mathit{Avg}\ '\mathit{Bd\text{-}lgth}]_{wt}\, x]\, '148]$
a_{11}	$\lambda w\lambda t\, \lambda x['= [['\mathit{Avg}\ '\mathit{Height}]_{wt}\, x]\, '75]$
a_{12}	$['\mathit{Biggest}\ ['\mathit{EU}\ ['\mathit{Feline}\ '\mathit{Predator}]]]$
a_{13}	$'\mathit{Has\text{-}mane}$
a_{14}	$\lambda w\lambda t\, \lambda x[['\leq ['\mathit{Bd\text{-}lgth}_{wt}\, x]\, '250] \wedge ['\geq ['\mathit{Bd\text{-}lght}_{wt}\, x]\, '170]]$
a_{15}	$['\mathit{Apex}\ '\mathit{Predator}]$
a_{16}	$\lambda w\lambda t\, \lambda x['= [['\mathit{Avg}\ '\mathit{Height}]_{wt}\, x]\, '117]$
a_{17}	$'\mathit{Pantherinae}$
a_{18}	$['\mathit{Significant}\ '\mathit{Sex\text{-}Dimorph}]$

sentences by the language of TIL constructions. TIL constructions are inputs for the supervised machine learning algorithm. In the next step, the FCA data mining method is applied on explications to obtain formal concepts. Combining the properties and attribute values provided by the user and with results of FCA, our method offers appropriate concepts which fall under properties and attributes' values provided by the user. The method is demonstrated by an example with 6 explications of different feline predators.

Acknowledgements. This research has been supported by Grant of SGS No. SP2021/87, VSB-Technical University of Ostrava, Czech Republic, "Application of Formal Methods in Knowledge Modelling and Software Engineering IV" and also supported by CZ.02.2.69/0.0/0.0/18_054/0014696 Rozvoj VaV kapacit Slezské univerzity v Opavě and SGS/11/2019 Rozvoj metod teoretické a aplikované informatiky.

References

1. Menšík, M., Duží, M., Albert, A., Patschka, V., Pajr, M., "Machine learning using TIL". In Frontiers in Artificial Intelligence and Applications, Amsterdam: IOS Press, Vol. 321, pp. 344-362, DOI: 10.3233/FAIA200024

2. Menšík, M., Duží, M., Albert, A., Patschka, V., Pajr, M.: ″Refining concepts by machine learning″. In Computación y Sistemas, Vol. 23, No. 3, 2019, pp. 943–958, doi: 10.13053/CyS-23-3-3242
3. Menšík, M., Duží, M., Albert, A., Patschka, V., Pajr, M.: ″Seeking relevant information sources″. In Informatics'2019, IEEE 15th International Scientific Conference on Informatics, Poprad, Slovakia, 2019, pp. 271-276.
4. Carnap, R: Meaning and Necessity: ″A Study in Semantics and Modal Logic″, Chicago, University of Chicago Press, 1964.
5. Duží, M., Jespersen, B.,Materna, P.: ″Procedural Semantics for Hyperintensional Logic″. Foundations and Applications of Transparent Intensional Logic.″, Berlin: Springer, 2010.
6. Menšík, M., Albert, A., Patschka, V.: ″Using FCA for seeking relevant information sources″. In Recent Advances in Slavonic Natural Language Processing, Volume 2020, pp. 47-54, 2020, ISBN 978-802631600-8
7. Russell, S.J.,Norvig, P.: ″Artificial intelligence: a modern approach.″, 2nd ed. Harlow, Pearson Education, 2014. ISBN 978-1-29202-420-2.
8. Mitchell, T.M.: ″Machine learning″, New York: McGraw-Hill, 1997. ISBN 00-704-2807-7.
9. Poole, D.L., Mackworth, A.K.: ″Artificial intelligence: foundations of computational agents. 2nd pub.″, Cambridge: Cambridge University Press, 2010, ISBN 978-0-521-51900-7.
10. Winston, P. H.: ″Artificial Intelligence″. 3rd ed., Mass.: Addison-Wesley Pub. Co., 1992. ISBN 02-015-3377-4.
11. Albert, A., Duží, M., Menšík, M.,Patschka, V.,Pajr, M.: ″Search for appropriate textual information sources″. In Frontiers in Artificial Intelligence and Applications, Amsterdam: IOS Press, Vol. 333, pp. 227-246, DOI: 10.3233/FAIA200832
12. Tichý, P.: ″The Foundations of Frege's Logic.″, Berlin, New York: De Gruyter, 1988.
13. Ganter, B., Wille, R.: ″Formal Concept Analysis: Mathematical Foundations″. 1st ed., Berlin: Springer, 1999, ISBN 978-3-540-62771-5.

Evaluating Long Contexts in the Czech Answer Selection Task

Marek Medveď, Radoslav Sabol, and Aleš Horák

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech republic
`{xmedved1, xsabol, hales}@fi.muni.cz`

Abstract. In the search for the answer to an open-domain question, the size of the search window, or the answer context, can greatly influence the resulting determination of the answer. The presented paper offers a detailed evaluation of different sizes of the answer context in case of Czech question answering. We compare six different context types in four different lengths. The conclusion of the experiments is that prolonging the context can improve the precision for specific types but in general the best results are obtained with one-sentence contexts.

Keywords: Question answering · Answer selection · Answer context · Evaluation

1 Introduction

The longer the preceding answer context is, i.e. the more we know about the question subject in advance, the more precise and certain the sought answer is. At least, this is a common assumption for the way how people search for an answer. In the computer Question Answering (QA) task, the benefits of longer contexts has not yet been thoroughly evaluated.

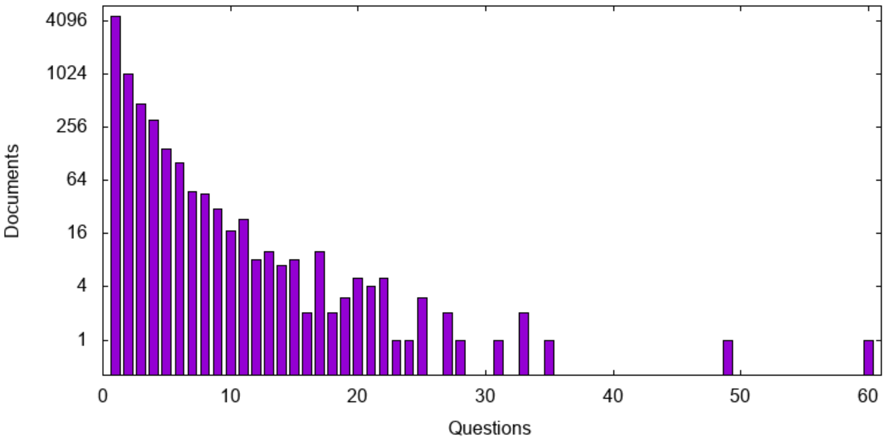
In this paper, we try to find the best answer context length experimentally. We evaluate and compare six different answer contexts setups each in four different lengths. The evaluation uses the Simple Question Answering Database (SQAD [4,6]) in version 3.1 and the results are compared with the answer selection task, i.e. the identification of the right document sentence which contains (or supports) the exact answer phrase.

To improve system performance, several related works examine context as a source of additional information. In [13], the authors used entities recognized in the question and a candidate concept and created an entity description based on Wiktionary definition. Afterwards, they employed this external entity descriptions to provide contextual information for knowledge understanding and achieved best results among non-generative models.

In [3], the authors modified BiDAF's [10] passage and question embedding processes to use the context information. According to their experiments, the context enhanced model outperformed the standard setup.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 61–69, 2021. © Tribun EU 2021

Fig. 1: Histogram of the numbers of questions per document.



2 Contexts in the SQAD Dataset

The latest Simple Question Answering Database (SQAD) introduced in [9,7] consists of 13 473 records created from 6 898 different Czech Wikipedia articles. Figure 1 displays the actual frequencies of the numbers of questions per documents.

The detailed statistics concerning the average length of sentence and question in the SQAD database are introduced in Table 1.

In the latest update of SQAD v. 3.1 introduced in [7], the database is enriched by contextual data in two main forms. Recurrent network (RNN) word embeddings are used as the first group of contexts that are added to each sentence during learning. They are formed by a sequence of individual word vectors to be concatenated with the candidate answer sequence during the learning process. The first sentence uses the text title as a context, because in many cases the title carries important information.

The second group of contexts is based on BERT-based sentence embeddings that are added into the model as one vector obtained from BERT model. In the

Table 1: SQAD text and question length statistics

Type	In tokens
Average text sentence length	20.18
Max text sentence length	205
Min text sentence length	1
Average question sentence length	8.22
Max question sentence length	43
Min question sentence length	1

experiments, several BERT based pre-trained modes have been used to encode the content of previous sentences.
The available context types that are used in the training phase are:

- RNN context types:
  - list of previous sentences (SENT)
  - list of link named entities[1] extracted from previous sentences (NE)
  - list of noun phrases extracted from previous sentences (PHR)
- transformer contexts types (transformer encodes previous sentences):
  - Czert [11] (CZT)
  - RobeCzech [12] (RC)
  - Slavic BERT [2] (SLB)

Each context type can be used in different sizes. Table 2 shows average context length in terms of tokens and items (item can be phrase, named entity or sentence) for RNN contexts. The transformer based only uses N vectors. The length determines how far back in text the context is calculated. The context length can have different impact to the final system performance (as we can see in Section 4). Additional features learned from context can therefore improve or degrade the final answer selection module performance used in the AQA [8] system.

Table 2: Average context lengths (in tokens) and average numbers of context items (e.g. number of different phrases) per the variable context window

Context type	context window (sentences)	average length of context in tokens	average number of context items
NE	1	2.29	1.49
	2	4.48	2.97
	3	6.70	4.45
	4	8.93	5.93
	5	11.16	7.41
PHR	1	13.77	5.08
	2	27.71	10.22
	3	41.55	15.33
	4	55.42	20.45
	5	69.30	25.58
SENT	1	19.97	1.00
	2	40.12	2.00
	3	60.24	3.00
	4	80.41	4.00
	5	100.60	5.00

Table 3: Running times of experiments with respect to the context type and window

Context type	Time (h)			
Window Size	1	2	3	4
PHR	10.75	14.4	18.2	20.81
NER	9.82	10.32	10.81	12.18
SENT	11.32	13.8	18.09	20.48
Transformer	13.56	13.71	13.88	13.96

3 Experiments

The answer selection module performs a ranking task, where each sentence of a document obtains a score according to its semantic similarity to the question. The neural network input is a triplet of a *question*, a *candidate answer*, and its *context*. Both the question and the answer are represented as a sequence of 500-dimensional *word2vec* word embeddings, while the context representation depends on the current context type as described in Section 2.

The first step utilizes a *Bidirectional Gated Recurrent Unit* (*BiGRU*) network to re-encode both the question and answer sequences into a hidden representation where their position in the sequence enriches each token. For RNN contexts, the same BiGRU layer is used to transform them into their hidden representation. However, a separate BiGRU has to be used instead for the transformer contexts, as the sequences are derived from a different language model. In both cases, the resulting hidden context vectors are concatenated to the candidate answer.

The following process involves an attention layer that assigns an importance score to each question token according to its importance in the answer and vice versa. This process also applies to both transformer and RNN contexts at the tail of the answer sequence (for example, an importance score can be assigned for the entire previous sentence vector in the transformer context).

The created attention vectors are multiplied with their corresponding hidden sequence. They result in two equally sized vectors, where their cosine similarity is the final ranking for the input triplet.

The SQAD dataset is partitioned into train/validation/test sets in the ratio 60:10:30. The partitions are balanced with regards to the ratio of question and

Table 4: The best hyperparameter values for various context types

Context Type	BiGRU Hidden Size	Learning Rate	Dropout
SENT RNN context	380	0.0004	0.4
PHR RNN context	380	0.0002	0.4
NER RNN context	320	0.0006	0.4
SENT Transformer ctx	480	0.0007	0.2

---

[1] See [5] for details about the specific named entity recognition technique.

Table 5: Mean average precision for each context type and context window size

Context type	Mean Average Precision							
Window Size	1		2		3		4	
MAP	S	M	S	M	S	M	S	M
PHR	*82.24*	*84.92*	82.23	84.98	80.56	83.41	80.55	83.31
NER	82.58	85.3	82.16	84.94	*82.71*	*85.53*	82.4	85.04
SENT	*81.9*	*84.76*	80.9	83.39	79.31	82.2	78.54	81.56
CZERT	**83.39**	**85.79**	82.71	85.38	82.76	85.36	82.78	85.35
ROBECZECH	*82.75*	85.29	82.46	85.05	82.69	*85.44*	82.56	85.14
SLAVIC_BERT	83.05	85.59	*83.19*	*85.91*	82.74	85.49	82.88	85.55

answer types. The training partition contains 8,059 records and is used to optimize the weights of the model. The validation set has 1,401 records and is used for an unbiased evaluation and early stopping (models are trained on 25 epochs, but the epoch with the best validation accuracy will be chosen as the result). The test set contains 4,013 records and is used for the final evaluation of the model.

We will refer to the number of preceding sentences from which the context is derived as the context *window size*. The primary goal of the following experiments is to determine the most optimal context window for each context type, and compare their performances. For this purpose, a window size from 1 to 4 is used for each type of context presented in Section 2. Larger context windows (`PHR_5` or `SENT_5`) could not be realized due to the technical limitations of the GPU. Each of the setups is repeated three times where the resulting mean average precision (MAP) score is recorded as the result of all runs.

4 Results and Discussion

The experiments were performed on Metacentrum adan clusters and were accelerated using the NVIDIA Tesla T4 graphics cards. Table 3 shows differences

Table 6: Best models per question type with different context types

Question type	Non context MAP in %	best context	window	best MAP in %	worst context	window	worst MAP in %
VERB_PHRASE	82.64	NE	3	**83.63**	SENT	4	76.71
ENTITY	79.40	SLB	1	**81.62**	SENT	4	75.47
NUMERIC	78.50	NE	1	**79.79**	SENT	4	72.95
ADJ_PHRASE.	83.89	SLB	1	**84.19**	SENT	4	79.53
CLAUSE	74.82	SLB	2	**75.78**	SENT	4	66.19
DATETIME	84.52	CZT	1	**84.80**	SENT	3	79.93
LOCATION	83.13	CZT	1	**86.61**	SENT	4	81.83
PERSON	81.33	CZT	1	**85.17**	SENT	3	81.59
ABBREVIATION	91.75	NE	4	**94.16**	SENT	2	90.03

Question:

Kolik sportovců se zúčastnilo XXVIII. letních olympijských her 2004 v Aténách?

[*How many athletes participated in the XXVIII-th Summer Olympic games in 2004 in Athens?*]

Answer from non-context model:

Her se zúčastnilo 202 zemí.

[*202 countries took part in the games.*]

Answer from the NER context model (window size of one sentence)

Účastnilo se jich 10625 sportovců z 201 zemí světa.

[*10625 athletes from 201 countries took part in them.*]

1st context item

letní olympijské hry

[*Summer Olympic games*]

2nd context item

Athénách

[*Athens*]

Fig. 2: An example answer where the NER context improved the system performance (record 000252)

in running times for various types of context. We can observe that in RNN contexts, the running time increases substantially with the increasing context window. For the transformer context, the running times are overall longer due to the additional BiGRU layer, which brings more parameters to optimize for the model. However, the increase in running times w.r.t window size is minimal as these contexts have more compact representations than the RNN ones.

The hyperparameters of the model were optimized semi-automatically using the Optuna hyperparameter optimization framework [1]. The original hyperparameter values from [7] have been used with increased context sizes. The list of the parameter setups per context can be seen in Table 4.

Table 5 presents the results for each context type and window size. The MAP scores in the *S* columns refer to the version where each record assumes only one single correct answer in the document, while *M* refers to the version where any sentence containing the exact answer is a correct answer, i.e. multiple correct answer sentences are allowed. The best result of each row is in italic, while the best result globally is in bold font. For the `PHR` and `SENT` contexts, the performance gradually degrades with the increasing context window. The decrease is due to a large number of tokens in the context, making it more difficult for the model to capture the dependencies of the sequence items. The `NER` context is more compact and produces slightly better results for the window size 3.

For the transformer contexts, a slight improvement in accuracy with the `RobeCzech` model and window size of 2 are recorded. Otherwise, the window size of 1 results in the best performance. Overall, the best setup uses the `Czert` transformer context with window size 1 and achieves the MAP score of 83.39 % in the single answer setup and 85.79 % in the multiple answers setup.

Question:

Je Jeruzalém jedno z nejstarších měst na světě?

[*Is Jerusalem one of the oldest cities in the world?*]

Answer from the non-context model:

Historie města sahá až do 4. tisíciletí př. n. l. a činí tak z Jeruzaléma jedno z nejstarších měst na světě.

[*The history of the city dates back to the 4-th millennium BC and makes Jerusalem one of the oldest cities in the world.*]

Answer from the SENT context model (window size of 4 previous sentences)

Nachází se v něm však také množství významných starověkých křesťanských míst a je považováno za třetí nejsvětější místo islámu.

[*However, there us also located a number of important ancient Christian sites and is considered the third holiest site in Islam.*]

1st context item

Jeruzalém se nachází v Judských horách na hranici úmoří Středozemního a Mrtvého moře na okraji Judské pouště.

[*Jerusalem is located in the Judean Mountains on the border of the Mediterranean and the Dead Sea, on the edge of the Judean Desert.*]

2nd context item

Současný Jeruzalém se rozrůstá daleko za hranicemi Starého Města.

[*Today's Jerusalem is growing far beyond the Old City.*]

3rd context item

Historie města sahá až do 4. tisíciletí př. n. l. a činí tak z Jeruzaléma jedno z nejstarších měst na světě.

[*The history of the city dates back to the 4-th millennium BC and makes Jerusalem one of the oldest cities in the world.*]

4th context item

Jeruzalém je nejsvětějším místem judaismu a duchovním centrem židovského národa.

[*Jerusalem is the holiest site of Judaism and the spiritual center of the Jewish nation.*]

Fig. 3: An example answer where longer sentence context degraded the system performance (record 009720)

We have also evaluated the answer selection module performance (mean average precision – MAP) with the new context types in relation to different question types. Table 6 reveals a significant improvement in the module performance when supplying some context to the training phase. A comparison among the question type results shows that two transformer contexts and one RNN context outperform the other context types. While also here for most question types the shorter context windows win, the NE model achieves the best performance for *verb phrase* with window size 3 and for *abbreviations* with window size 4. Presumably, these question types are frequently explained in longer texts than the other types of questions. The SENT context with large window sizes significantly decreases the module performance.

Examination of the results shows why the named entities (NE) context improves the module performance. Figure 2 shows that named entities extracted from previous sentences provide the important additional information that helps the system to choose the right sentence. The entities of *Summer Olympic games* and *Athens* resolve the anaphora appearing in the correct answer and finds the important antecedents contained in the question.

The Slavic BERT and Czert context types do not offer such explainable representation of the context. Overall, their dense sentence representation allows to encode the important aspects of the sentence even slightly better than the NE context even though they do not specifically point at the important pieces of information in the context.

On the other hand if we look on the performance of the SENT context model with window size of 4 previous sentences, we can see significant decrease in the final module performance. A specific example is presented in Figure 3, where the resulting sentence context is too long. This finally confuses the model with too much additional information. Also the context of the selected sentence contains the correct sentence which should have been selected as the correct answer.

5 Conclusions

In the paper, we have evaluated the assets of using several answer contexts in varying context lengths to solve the answer selection task. The results reveal that for specific question types, such as verb phrases or abbreviations, longer contexts in the form of important entities improve the performance. In all cases, the context representation is better than a model with no context information. However, in prevailing number of cases, the best context size uses just one preceding sentence as the source of context information and with widening the context window the benefits of using the context diminish and actually degrade the performance.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631 (2019)
2. Arkhipov, M., Trofimova, M., Kuratov, Y., Sorokin, A.: Tuning multilingual transformers for language-specific named entity recognition. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing. pp. 89–93. Association for Computational Linguistics, Florence, Italy (Aug 2019).

https://doi.org/10.18653/v1/W19-3712, `https://www.aclweb.org/anthology/W19-3712`
3. Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W., Choi, Y., Liang, P., Zettlemoyer, L.: Quac : Question answering in context. CoRR **abs/1808.07036** (2018), `http://arxiv.org/abs/1808.07036`
4. Horák, A., Medveď, M.: SQAD: Simple question answering database. In: Eighth Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 121–128. Tribun EU, Brno (2014)
5. Medved, M., Sabol, R., Horák, A.: Efficient Management and Optimization of Very Large Machine Learning Dataset for Question Answering. In: RASLAN 2020. pp. 23–34 (2020)
6. Medveď, M., Sabol, R., Horák, A.: Employing Sentence Context in Czech Answer Selection. In: International Conference on Text, Speech, and Dialogue, TSD 2020. pp. 112–121. Springer (2020)
7. Medveď, M., Sabol, R., Horák, A.: Comparing RNN and Transformer Context Representations in the Czech Answer Selection Task. In: Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022). SCITEPRESS, Setúbal, Portugal (2022), in print
8. Medveď, M., Horák, A.: Sentence and word embedding employed in open question-answering. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018). pp. 486–492. SCITEPRESS - Science and Technology Publications, Setúbal, Portugal (2018)
9. Sabol, R., Medveď, M., Horák, A.: Czech question answering with extended SQAD v3.0 benchmark dataset. In: Proceedings of the 13th Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2019. pp. 99–108. Tribun EU (2019)
10. Seo, M.J., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. CoRR **abs/1611.01603** (2016), `http://arxiv.org/abs/1611.01603`
11. Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., Konopík, M.: Czert – Czech BERT-like Model for Language Representation. arXiv preprint arXiv:2103.13031 (2021)
12. Straka, M., Náplava, J., Straková, J., Samuel, D.: RobeCzech base (2021), `http://hdl.handle.net/11234/1-3691`, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
13. Xu, Y., Zhu, C., Xu, R., Liu, Y., Zeng, M., Huang, X.: Fusing context into knowledge graph for commonsense reasoning. CoRR **abs/2012.04808** (2020), `https://arxiv.org/abs/2012.04808`

Questions and Answers on Dynamic Activities of Agents

Marie Duží

VSB-Technical University of Ostrava,
Department of Computer Science FEI,
17. listopadu 15, 708 33 Ostrava, Czech Republic
marie.duzi@vsb.cz

Abstract. In a multiagent and multi-cultural world, the fine-grained analysis of agents' dynamic behaviour, i.e. of their activities, is essential. Dynamic activities are actions that are characterised by an agent who executes the action and by other participants of the action. Wh-questions on the participants of the actions pose a difficult particular challenge because the variability of the types of possible answers to such questions is huge. To deal with the problem, we proposed the classification of the participants of activities that is inspired by linguistic classification of verb valency verbs. The application of these results to the analysis of processes and events and to questioning and answering about these activities is a novelty of the paper.

Keywords: Activity · Communication of agents · Transparent Intensional Logic · Wh-questions and answers

1 Introduction

The primary goal of this paper is to *logically* analyse *processes* and *activities* so that the agents in a multiagent and multicultural world can ask on the participants of such activities. To this end, we have defined different kinds of possible participants of an activity; this classification is inspired by linguistic verb valency frames. Hence, different kinds of Wh-questions and plausible answers can be derived, as each specialised subtype of a Wh-question conveys specific information for an agent on how and where to seek the corresponding direct answer. In addition, by applying TIL deduction system, the agents can infer even more detailed answers, if needed. Thus, we wish to provide not only direct answers extracted from natural-language texts or agents' knowledge bases just by keywords; rather, we also want to derive logical consequences of such answers. Currently, the need of a hyperintensional approach to natural-language processing is broadly recognised. For these reasons, we vote for Transparent Intensional Logic (TIL) as our background theory.[1] Duží and Fait introduce in [7] Genzen's system of natural deduction adjusted for TIL and

---

[1] See, for instance, [5], [15], [14], [8].

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 71–81, 2021. © Tribun EU 2021

natural-language processing. The analysis of Wh-questions results into λ-terms with a free variable x ranging over entities of type α, which is the type of a possible direct answer. The system provides answers by suitable substitutions of the α-entities extracted from input sentences, the constituents of which match a given λ-term. It also makes it possible to derive as an answer even more information by applying the semantic rules rooted in the rich semantics of a natural language. In particular, the agents can make use of the relations of requisites and pre-requisites between intensions.

The rest of the paper is organised as follows. Section 2 introduces the basic principles of Transparent Intensional Logic (TIL) that is my background logical system. Section 3 introduces the main results of this paper; it deals with the TIL technique of answering Wh-questions, and concentrates in particular on the dynamic activities of agents. Concluding remarks can be found in Section 4.

2 Basic principles of TIL

Pavel Tichý, the founder of Transparent Intensional Logic (TIL) was inspired by Frege's semantic triangle.[2] However, while Frege did not define the sense of an expression but only characterised it as the 'mode of presentation', Tichý ([21], [22]) defined the sense of an expression, i.e. its *meaning*, as an abstract, algorithmically structured *procedure* that produces the object denoted by the expression, or in rigorously defined cases fails to produce a denotation if there is none.[3]

Tichý in [25] defined six kinds of meaning procedures and called them *constructions*. There are two kinds of atomic constructions that present input objects to be operated on by molecular constructions. They are *Trivialization* and *variables*. Trivialisation of an object X presents the object X without the mediation of any other procedures. Using the terminology of programming languages, the Trivialisation of X, denoted by '${}^0X$', is just a pointer or reference to X. Trivialization can present an object of any type, even another construction C. Hence, if C is a construction, ${}^0C$ is said to present the construction C, whereby C occurs *hyperintensionally*, i.e. in the *non-executed* mode. Variables produce objects dependently on valuations; they are said to *v-construct*. The execution of a Trivialisation or a variable never fails to produce an object. However, since TIL is a logic of *partial functions*, the execution of some of the molecular constructions can fail to present an object of the type they are typed to produce. When this happens, we say that a given construction is *v-improper*. This concerns in particular one of the molecular constructions, namely *Composition*, $[XX_1 \dots X_m]$. It is the very *procedure of applying a function f* produced by X (if any) to the tuple argument $\langle a_1 \dots a_m \rangle$ (if any) produced by the procedures $X_1, \dots, X_m$. A Composition is v-improper as soon as f is a partial function not defined at its tuple argument, or if one or more of its constituents $X, X_1, \dots, X_m$ are v-improper. Another molecular construction is λ-*Closure*, $[\lambda x_1 \dots x_m X]$. It is the very *procedure of producing a function* with the values v-produced by the procedure X, by abstracting over the values of the variables $x_1, \dots, x_m$ to provide functional arguments. No Closure is v-improper for any valuation v, as a Closure always v-constructs a function (which may be, in an extreme case, a degenerate function undefined at all its arguments). Each construction C can occur not only in execution mode designed to produce an object (if any) but also as an object in its own right on which other (higher-order) constructions operate. The Trivialisation of C causes C to occur just presented as an argument, as mentioned above. Yet sometimes, we need to cancel the effect of Trivialisation and trade the mode of C for execution mode. Double Execution, ${}^2C$, does just that; it executes C twice over. If C v-constructs a construction D that in turn v-constructs an entity E, then ${}^2C$ v-constructs E. Otherwise, ${}^2C$ is v-improper. Hence, the following ^{20}–Elimination rule is valid; for any construction C, ${}^{20}C=C$.

---

[2] See [25].

[3] A similar philosophy of meaning as a 'generalized algorithm' can be found in [18]; this conception has been further developed by Loukanova, see [17].

TIL is a typed λ-calculus. Hence, each entity, even a construction, receives the type to which it belongs. The inductive definition of the *ramified hierarchy of types*, as any inductive definition, consists of a base, inductive steps and the closure. For the purposes of natural-language analysis, we are usually assuming the following *base of ground types*: o (the set of truth-values true $\mathbf{T}$ and false $\mathbf{F}$), ι (the set of individuals, i.e. the universe of discourse),[4] τ (times or real numbers) and ω (possible worlds). From these types of *non-procedural* objects, on the ground level of types of order 1, partial functions of type $(\alpha_1 \dots \alpha_m)$ are defined inductively. Second, constructions of order n are defined as those procedures that produce objects of a type of order m, where $1 \leq m \leq n$. However, these constructions form a higher-order type $*_n$, which is a type of order $n+1$. Finally, partial functions belonging to a type of order $n+1$ are of type $(\alpha\alpha_1 \dots \alpha_m)$, where at least one of the types $\alpha, \alpha_1, \dots, \alpha_m$ is equal to $*_n$.

Empirical expressions denote *empirical conditions*, which may or may not be satisfied at the world/time pair selected as points of evaluation. These empirical conditions are modelled as (PWS-)*intensions*. Intensions are entities of type $((\alpha\tau)\omega)$, or $\alpha_{\tau\omega}$, for short. *Extensional entities* are entities of a type α where $\alpha \neq (\beta\omega)$ for any type β.

Notational conventions. The outermost brackets of Closures are omitted whenever no confusion can arise. Furthermore, 'X/α' means that an object X is (a member) of type α, and '$X \to \alpha$' means that X is typed to v-construct an object of type α. Throughout, it holds that the variables $w \to \omega$ and $t \to \tau$. If $C \to \alpha_{\tau\omega}$ then the frequently used Composition $[[Cw]t]$, which is the *extensionalization* of the α-intension v-constructed by C, is encoded as 'C_{wt}'.

3 Wh-questions and answers

3.1 Technique of answering Wh-questions

From the logical point of view, *empirical questions* denote α-intensions and the direct answer to such a question is the value of type α of this intension in the actual world and time.[5] Hence, the type of possible answer dictates the type of empirical question. Empirical Yes-No questions denote *propositions* of type $o_{\tau\omega}$, where o is the type of truth-values.[6] However, the variety of possible answers to Wh-questions is much greater depending on the type α of an α-intension. For instance, one can ask for the value of an *individual office* (or role) of type $\iota_{\tau\omega}$, like "Who is Miss World 2021"? A possible answer to such a question is a unique individual (an object of type ι who happens to play a given role). Another frequent type of intensions is the *property of individuals*, an object of type $(o\iota)_{\tau\omega}$. For instance, the direct answer to the question "Which Czech ladies are among the first fifty players in WTA ranking singles?" should convey a set (of type $(o\iota)$) of individuals. Currently (written 2021/11/12), they are Barbora Krejčíková, Karolina Plíšková, Petra Kvitová, Karolína Muchová, Marketa Vondroušová. Hence, the question denotes a property of individuals, namely that of being a female Czech tennis player among the first fifty in WTA ranking singles. One can also ask for the value of an attribute at an argument like the salary of somebody. The possible answer to the question "What is John's salary?" is some number of type τ. Hence, the question denotes a magnitude of type $\tau_{\tau\omega}$.

---

[4] We assume that the universe of discourse is a multi-valued set consisting of at least two elements, though we leave aside the cardinality of this basic type.

Duží and Fait in [7] introduce a useful logical technique of answering Wh-questions. The answers are obtained by suitable substitutions, i.e. unifications known from the general resolution method. For a simple example, assume that in an agent's knowledge base, there are these formalised sentences.

(1) $\lambda w \lambda t\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ {}^0\mathit{Barty}] = {}^0 1]$
(2) $\lambda w \lambda t\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ {}^0\mathit{Sabalenka}] = {}^0 2]$
(3) $\lambda w \lambda t\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ {}^0\mathit{Krejcikova}] = {}^0 3]$
(4) $\lambda w \lambda t\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ {}^0\mathit{Pliskova}] = {}^0 4]$
(5) $\lambda w \lambda t\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ {}^0\mathit{Muguruza}] = {}^0 5]$
And so on ...

The answer to the question "Who are the first three players in WTA tennis singles"?, i.e.

$$\lambda w \lambda t\, \lambda x\, [[{}^0\mathit{WTA\text{-}ranking}_{wt}\ x] \leq {}^0 3]$$

---

[5] Duží and Číhalová [6] distinguish between *direct* and *complete* answer to an empirical question. Direct answer is an object X of type α that is the value (in the world and time of evaluation) of the α-intension asked for, while complete answer is the proposition that the value of the asked intension is the object X. The authors deal with presuppositions of questions. Their main thesis is this. If a presupposition of a given question is not true, then there is no direct answer. Instead, a plausible complete answer is the negated presupposition.

[6] For details on TIL analysis of questions and answers see [9, §3.6].

is derived like this. Question (raised in a given w and t)[7]

(6) $\lambda x\, [[{}^0 WTA\text{-}ranking_{wt}\, x] \leq {}^0 3]$

(7) $[[{}^0 WTA\text{-}ranking_{wt}\, x] \leq {}^0 3]$ 6, λ-E

To answer the question, the algorithm searches a given knowledge base for those sentences the constituents of which match with (7). In addition, basic algebraic operations can be applied. Thus, the first matching sentence is (1), as $1 \leq 3$. By substituting ${}^0 Barty$ for the variable x, we obtain the answer $x = {}^0 Barty$. Since the question concerns the set of three individuals, the algorithm searches for another matching sentence, which corresponds answering the question "Who else"? In the exactly same way, the answers $x = {}^0 Sabalenka$ and $x = {}^0 Krejcikova$ are conveyed.

Though WTA tennis ranking is changing frequently, as these are empirical facts, from the point of view of dynamic behaviour of agents, the analysis of their activities is the most important issue.

3.2 Dynamic activities

A large number of Wh-questions concerns the participants of activities. Yet, these participants often belong to just one logical type, which is too coarse-grained. We need more detailed classification of their types. Linguistic classifications of Wh-questions are mostly based on the types of question pronouns, i.e. descriptors of interrogative sentences, for example, why, where, how, etc.[8] Descriptors refer to objects of various types. In other words, Wh-questions can ask for time, reason, manner, individuals, the definition of something, etc. Hence, a significant amount of different types of queries belong under the umbrella of Wh-questions.

Our specification of *activities* is based on the linguistic theory of *verb valency frames* and on their logical analysis.[9] From the logical point of view, we deal with the verb phrases as denoting a *function* that is applied to its arguments. The number of arguments is controlled by the content verb valency.[10] Verb valency frames determine the obligatory and facultative arguments, i.e. thematic roles of a given verb, together with their types. Linguists have developed many classifications based on verb valency frames, for instance, VALLEX or Verba Lex.[11] Sowa [19] distinguishes several types of thematic roles, for instance, Agent, Beneficiary, Destination, Duration, Effector, Experiencer, Instrument, Location, Matter, Patient and so on (ibid., pp. 508-510). Thematic role or the type of participant expresses the role that a noun phrase plays with respect to the activity described by a governing verb. From the viewpoint of logic, it is the relation between two entities where one of them is an activity (expressed by the verb), and the other is a participant (expressed mostly by a noun, adverb or adjective). The number and the categories of participants depend on the respective domain of interest and the functions of the system of agents. Being inspired by these ideas, we primarily use the following frequent kinds of participants:

---

[7] When applying a proof in TIL, the first steps eliminate the left-most $\lambda w \lambda t$, which corresponds to two β-conversions. They apply the empirical assumptions to the world w and time t of evaluation to obtain a truth-value. Similarly, Wh-question transforms into a procedure producing an object of type α. For details, see [7].

[8] See, for instance, [11] and [27].

[9] For the linguistic theory of verb valency frames, see [13]; see also [2] for the proposal of an ontology of events based on the theory of verb valency frames.

[10] For details, see [3].

[11] See, for instance [16] and [12].

Pat; object affected by the activity
Ben; beneficient (somebody who has a benefit from the activity)
Manner; the manner of the activity execution (measure, speed etc.)
Inst; instrument
Time; when the activity takes place
Time1; when the activity begins
Time2; when the activity ends
Loc; the place of activity
Dir1; the direction of activity, from where
Dir2; the direction of activity, which way
Dir3; the direction of activity, where to

If needed, other kinds of attributes can be specified; we only must keep the selected keywords fixed.

Questions concerning activities can be on the process itself (what is going on?), questions on the primary agent (who or what is doing so and so) and on other participants of a given activity. For instance, assume we have the sentence "John (the agent) is going (the activity) to Brussel (Dir3) by car (Inst) at an average speed of 60 miles per hour (Man)." Then we can ask, "What is John doing?", "Who is going to Brussel?", "How quickly does John go to Brussel?", etc. Our classification enables an agent to look for sentences that might provide a plausible answer at an appropriate component of the agent's knowledge base provided this piece of knowledge is there, or ask their fellow agents, or look for the answer in the huge amount of natural-language texts available.

The basic idea of logical analysis of activities and events is due to Tichý [24]. Its adjustment and simplification have been introduced in [1]). Tichý draws a distinction between *episodic* and *attributive* verbs. Attributive verbs ascribe properties to individuals. Their structure is usually a copula followed by an adjective or noun; for instance, 'is happy', 'is red', 'looks speedy', 'is a student' are attributive verbs. On the other hand, episodic verbs express actions performed by entities. For instance, if John is getting up, it does not suffice to analyse this activity by assigning the property of getting up to John. Instead, John is *doing* the activity of getting up, and one can ask, for instance, "When does John get up?".

Each activity can be specified by a verb *Do*, and by *Who* (the actor), *What* (the activity that is being done), possibly with the attributes of the activity like objects to be operated on, resources, etc. Using a general place holder π for the type of activity and $\alpha^{Part\text{-}i}$ for an attribute/participant of a kind *Part-i*, the type of

Do is $(o\iota\pi)_{\tau\omega}$, and the assignment of participants to the activity is then an entity *Ass* of type $(o\pi\alpha^{Part\text{-}i})_{\tau\omega}$. To simplify the notation and make the formulas easier to read, we will use '${}^0X^{Part\text{-}i}$' instead of '$[{}^0Part\text{-}i\;{}^0X]$' to signify that X belongs to the class of participants *Part-i*. Thus, we obtain a general pattern for analysing an activity $P \to \pi$ with the actor $a \to \iota$ and participants $X_1^{Part\text{-}1}, \ldots, X_n^{Part\text{-}n}$.

$$\lambda w \lambda t\, [[{}^0Do_{wt}\; a\; P] \wedge [{}^0Ass_{wt}\; P\; {}^0X_1^{Part\text{-}1}] \wedge \ldots \wedge [{}^0Ass_{wt}\; P\; {}^0X_n^{Part\text{-}n}]]$$

For instance, the analysis of the sentence "*John builds a house in Bali*" comes down to this construction.

$$\lambda w \lambda t\, [[{}^0Do_{wt}\; {}^0John\; {}^0Build] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0House^{Pat}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Bali^{Loc}]]$$

It may happen that in another time John would build a house in Rome. Then we have

$$\lambda w \lambda t\, [[{}^0Do_{wt}\; {}^0John\; {}^0Build] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0House^{Pat}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Rome^{Loc}]]$$

For this reason, the relation *Ass* between an activity and its participant is the relation-in-intension rather than in extension.

If there are two or more actors of the activity, we apply the relation-in-intension $Do/(o\iota \ldots \iota\pi)_{\tau\omega}$. For instance, the sentence "*John and Tom build a house in Rome*" is furnished with this analysis.

$$\begin{gathered}\lambda w \lambda t\, [[{}^0Do_{wt}\; {}^0John\; {}^0Tom\; {}^0Build] \wedge \\ [{}^0Ass_{wt}\; {}^0Build\; {}^0House^{Pat}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Rome^{Loc}]]\end{gathered}$$

If an agent b has in their ontology the specification of all the possible participants of activity, and b obtains an incomplete message concerning the activity, then b can ask his fellow agents for completing their pieces of knowledge. For instance, when receiving the first message about John's building a house in Bali, the agent can ask *when* and for *whom* does John build the house. To this end, we use variables $when \to (o\tau)$ and $whom \to \iota$, the valuation of which would be the answer. The content of the query is then this.

$$\begin{gathered}\lambda w \lambda t\, \lambda when \lambda whom\, [[{}^0Do_{wt}\; {}^0John\; {}^0Build] \wedge \\ [{}^0Ass_{wt}\; {}^0Build\; {}^0House^{Pat}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Bali^{Loc}] \wedge \\ [{}^0Ass_{wt}\; {}^0Build\; when^{Time}] \wedge [{}^0Ass_{wt}\; {}^0Build\; whom^{Ben}]]\end{gathered}$$

A possible direct answer to agent b is $when = {}^0November\text{-}2021$, $whom = {}^0Marie$.

Another advantage of this approach is this. Since in TIL we have two modal parameters, time and possible worlds, we can easily analyse the activities executed in *past* or *future* and model dynamic behaviour and reasoning of agents. For instance, the question "*When did John build a house in Bali for Marie*"? receives this analysis.

$$\begin{gathered}\lambda w \lambda t\, \lambda when\, \exists t' [[[{}^0Do_{wt'}\; {}^0John\; {}^0Build]\; \wedge [t' \leq t]] \wedge \\ [{}^0Ass_{wt}\; {}^0Build\; {}^0House^{Pat}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Bali^{Loc}] \wedge \\ [{}^0Ass_{wt}\; {}^0Build\; when^{Time}] \wedge [{}^0Ass_{wt}\; {}^0Build\; {}^0Marie^{Ben}]]\end{gathered}$$

The situation gets more complicated if a sentence in past or future comes with a *time reference* T when this or that happened or will happen. In such a case, the sentence is associated with a *presupposition* that the current time t is in the proper relation with respect to the reference time T. Roughly, it means that for sentences in future t comes before the reference time T, while for sentences in past t comes after T; if it is not so, then the proposition has a truth-value gap. Moreover, the sentence can also convey information on the *frequency* of the activity to be executed in the reference time T like twice, always, all the time since, for the whole year. Duží in [4] demonstrates the method of a fine-grained analysis of such sentences in past and future with a reference time interval T. In the paper, a general analytic schema for sentences that come associated with a presupposition is presented. To this end, the author utilises a strict definition of the *If-then-else-fail* function that complies with the compositionality constraint.

For instance, the truth conditions of the sentence *"John has built a house in Bali in 2020"* presuppose that the current time t in which the truth conditions are being evaluated comes after the end of 2020. If it is not so, the sentence has *no truth value*. Thus, we have

$$\lambda w \lambda t\, [\mathit{If}\, [t \geq_\tau {}^0 2020]\ \mathit{then}$$

$$[\exists t'\, [[{}^0\mathit{Do}_{wt'}\ {}^0\mathit{John}\ {}^0\mathit{Build}] \wedge [{}^0 2020\, t']] \wedge$$
$$[[{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ {}^0\mathit{House}^{\mathit{Pat}}] \wedge [{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ {}^0\mathit{Bali}^{\mathit{Loc}}] \wedge [{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ {}^0 2020^{\mathit{Time}}]]]$$

$$\mathit{else\ fail}]$$

Additional types. $2020/(o\tau)$; $\geq_\tau /(o\tau(o\tau))$: the relation between the evaluation time t and time interval of the year 2020 such that t comes after the end of the year 2020.[12] The path with the statement 'else fail' means that the denoted proposition evaluates to *no truth value*.

However, if an agent asks without time reference, "*When did John build a house in Bali*?", then the test on the temporal presupposition validity is not applied, of course. Thus, we have $(\mathit{when} \rightarrow (o\tau))$

$$\lambda w \lambda t\, \lambda \mathit{when}\, [\exists t' [[{}^0\mathit{Do}_{wt'}\ {}^0\mathit{John}\ {}^0\mathit{Build}] \wedge [t' \leq t]] \wedge$$
$$[[{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ {}^0\mathit{House}^{\mathit{Pat}}] \wedge [{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ {}^0\mathit{Bali}^{\mathit{Loc}}] \wedge [{}^0\mathit{Ass}_{wt}\ {}^0\mathit{Build}\ \mathit{when}^{\mathit{Time}}]]]$$

By applying the above-described method of unification, the direct answer is $\mathit{when} = {}^0 2020$.

The method of analysis takes also account of the *frequency* of the activity to be executed in the reference time interval *In-Time*. The general analytic schema for sentences S in past tenses is this.

$$\lambda w \lambda t\, [{}^0\mathit{Past}_t\, [{}^0\mathit{Frequency}_w\, S]\ {}^0\mathit{In\text{-}Time}] =$$

$$\lambda w \lambda t\, \mathit{If}\, [{}^0\mathit{In\text{-}Time} \leq_\tau t]\ \mathit{then}\ [[{}^0\mathit{Frequency}_w\, S]\ {}^0\mathit{In\text{-}Time}]\ \mathit{else\,fail}$$

---

[12] More on dealing with time and calendars can be found in [10].

Here $\leq_\tau$ means that the reference interval *In-Time* comes before time t, or, in general, in a proper relation with respect to time t. Past receives the same type as Future (which is applied for sentences in future), that is $((o(o(o\tau))(o\tau))\tau)$; S is the proposition to be evaluated and *Frequency* of type $((o(o\tau))o_{\tau\omega}\,\omega)$ is the frequency of time intervals in which the proposition S takes the truth-value **T** in world w. The schema for sentences in future tenses differs only by applying the constituent *Future* instead of *Past*.[13]

If John often built houses in Bali since 2007, then by applying the above schema, we obtain this construction.

$$\lambda w \lambda t\, [{}^0\mathit{Past}_t\, [{}^0\mathit{Often}_w\, \lambda w \lambda t\, [[{}^0\mathit{Do}_{wt}\, {}^0\mathit{John}\, {}^0\mathit{Build}] \wedge \\ [{}^0\mathit{Ass}_{wt}\, {}^0\mathit{Build}\, {}^0\mathit{House}^{\mathit{Pat}}] \wedge [{}^0\mathit{Ass}_{wt}\, {}^0\mathit{Build}\, {}^0\mathit{Bali}^{\mathit{Loc}}]]]\, {}^0 2007]$$

The frequency modifier *Often* denotes a world-dependent function that takes a proposition $p \to o_{\tau\omega}$ to the class of those intervals $d \to (o\tau)$ which are contained in the chronology of p (i.e. $p_w \to (o\tau)$). Letting aside vagueness of the term 'often', be it twice or three times a year, if these intervals are frequent since 2007, the proposition is evaluated to **T**.

4 Conclusion

In this paper, I dealt with logical analysis of Wh-questions and its utilisation in intelligent communication and reasoning of agents in a multiagent world. I introduced logical analysis of Wh-questions and the way of their answering by applying Gentzen's natural deduction system adjusted to natural-language processing in TIL. I concentrated on the dynamic aspects of agents' reasoning, in particular questions on participants of activities specified in different tenses with reference time and frequency when this or that activity happened or will happen to be done.

Acknowledgements. This research was supported by the University of Oxford project 'New Horizons for Science and Religion in Central and Eastern Europe' funded by the John Templeton Foundation. The opinions expressed in the publication are those of the author(s) and do not necessarily reflect the view of the John Templeton Foundation. This research has been also supported by Grant of SGS No. SP2021/87, VŠB - Technical University of Ostrava, Czech Republic, "Application of Formal Methods in Knowledge Modelling and Software Engineering IV".

References

1. Číhalová, M., Duží, M., Menšík, M., Vích, L. (2011). Process ontology. In P. Sojka (ed.), *RASLAN 2010* (pp. 77-88). Brno: CNP MUNI.

---

[13] A detailed analysis of particular kinds of tenses can be found in [9, §2.5.2] or in [24].

2. Číhalová, M. (2016). Event ontology specification based on the theory of valency frames. In T. Welzer, H. Jaakkola, B. Thalheim, Y. Kiyoki, N. Yoshida (eds.) *Frontiers in Artificial Intelligence and Applications*, Information Modelling and Knowledge Bases XXVII, pp. 299-313, Amsterdam: IOS Press.
3. Dixon, R. M. W. (2000). A Typology of Causatives: Form, Syntax, and Meaning. In R. M. W. Dixon and A. Y. Aikhenvald (eds.), *Changing Valency: Case Studies in Transitivity*, pp. 30-41, New York, NY: Cambridge University Press.
4. Duží, M. (2010). Tenses and truth-conditions: a plea for if-then-else. In Peliš, M. (ed.) *The Logica Yearbook* 2009, pp. 63-80, London: College Publications.
5. Duží, M. (2019). If structured propositions are logical procedures then how are procedures individuated? *Synthese* special issue on the Unity of propositions, 196 (4), pp. 1249-1283.
6. Duží, M., Číhalová, M. (2015). Questions, answers and presuppositions. *Computación y Sistemas*, 19(4), pp. 647-659.
7. Duží, M., Fait, M. (2021). A hyperintensional theory of intelligent question answering in TIL. In: R. Loukanova (ed.), *Natural Language Processing in Artificial Intelligence* - NLPinAI 2020, Springer Series Studies in Computational Intelligence (SCI), vol. 939, pp. 69-104, Springer.
8. Duží, M., Fait, M., Menšík, M. (2017): Context Recognition for a Hyperintensional Inference Machine. In the AIP proceeding of ICNAAM 2016, *International Conference of Numerical Analysis and Applied Mathematics*, vol. 1863, Article No. 330004
9. Duží, M., Jespersen, B. Materna, P. (2010). *Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic*. Series Logic, Epistemology, and the Unity of Science, vol. 17. Berlin: Springer.
10. Duží, M., Macek, J. (2018). Analysis of time references in natural language by means of Transparent Intensional Logic. *Organon F*, 25(1), pp. 21-40.
11. Essberger, J. *WH-question words*. English Club. `http://www.englishclub.com/vocabulary/wh-question-words.htm`. Accessed January 1, 2021.
12. Hlaváčková, D., Horák, A. (2006). VerbaLex - New Comprehensive Lexicon of Verb Valencies for Czech. In *Computer Treatment of Slavic and East European Languages*, pp. 107-115, Bratislava, Slovakia: Slovenský národný korpus.
13. Horák, A. (1998). Verb Valency and Semantic Classification of Verbs. In *Proceedings of TSD'98*, pp. 61-66, Brno (CR): Masaryk University.
14. Jespersen, B. (2020). First among equals; co-hyperintensionality for structured propositions. *Synthese*, doi `https://doi.org/10.1007/s11229-020-02987-4`.
15. Jespersen, B., Duží, M. (2015). Introduction to the special issue on Hyperintensionality. *Synthese*, 192(3), pp. 525-534. DOI: 10.1007/s11229-015-0665-9
16. Lopatková, M., Žabokrtský, Z., Kettnerová, V. (2006). VALLEX 2.5. – Logical structure of the lexicon. `http://ufal.mff.cuni.cz/vallex/2.5/doc/structure-en.html.sec:frame`. Accessed October 1, 2021.
17. Loukanova, R. (2009). β-reduction and antecedent-anaphora relations in the language of acyclic recursion. In J. Cabestany et al. (eds.), IWANN 2009, Part I, pp. 496–503, vol. 5517 of *Lecture Notes in Computer Science*.
18. Moschovakis, Y. N. (2006). A logical calculus of meaning and synonymy. *Linguistics and Philosophy*, vol. 29, pp. 27–89.
19. Sowa, J. F. (2000). *Knowledge representation (logical, philosophical, and computational foundations)*. Pacific Grove, CA: Brooks Cole Publishing Co.
20. Sowa, J. F.: Thematic roles. `http://www.jfsowa.com/ontology/roles.htm`. Accessed January 1, 2021

21. Tichý, P. (1968). Smysl a procedura. *Filosofický časopis*, vol. 16, pp. 222-232. Reprinted in English as 'Sense and procedure' in (Tichý 2004: 77-92).
22. Tichý, P. (1969). Intensions in terms of Turing machines. *Studia Logica*, vol. 26, pp. 7-25. Reprinted in (Tichý 2004: 93-109).
23. Tichý, P. (1978). Questions, answers and logic. *American Philosophical Quartely*, vol. 15, pp. 275-284. Reprinted in (Tichý 2004: 293-304).
24. Tichý, P. (1980). The semantics of episodic verbs. *Theoretical Linguistics*, vol. 7, pp. 263-296. Reprinted in (Tichý 2004: 411–446).
25. Tichý, P. (1988). *The Foundations of Frege's Logic*, De Gruyter.
26. Tichý, P. (2004). *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.). Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press.
27. Types of Wh-Questions. Rochester Institute of Technology. `https://www.rit.edu/ntid/sea/processes/wh/grammatical/types`. Accessed January 1, 2021.

Conceptual Framework for Process Ontology

Martina Číhalová

Palacký University Olomouc,
Department of Philosophy,
Czech Republic
`martina.cihalova@upol.cz`

Abstract. The author compares different approaches to process and event conceptualization in this article in order to obtain basic concepts and their definitions on which the ontology of processes needs to be built. With an emphasis on the aspect of sharing of ontologies, the conceptual framework for process ontology is designed to be close to natural language and existing process or event ontologies and logical conceptualizations. In the natural language, each event is specified using some special type of verb as a component of the phrase describing the respective event. This type of verb is called an episodic verb according to Tichý's distinction between episodic and attributive verbs. The referent of episodic verbs is referred to as an activity in this article and it is the crucial concept of process ontology building. The specification of activities is driven by the linguistic theory of verb-valency frames.

Keywords: Process · Event · Ontology · Activity · Verb-valency frames

1 Introduction

The problem of conceptualization of processes concerns not only philosophy and logic but also computer science. This problem represents a challenge at present especially for the field of artificial intelligence where the reasoning of intelligent agents has temporal aspects and has to deal with changes in their environment. To obtain basic concepts for process ontology and their definitions, different approaches to process and event conceptualization are compared in section 2, namely well-known ontological languages such as Event Ontology, etc., or situation and event calculus. The article suggests that ontologies may be linguistically based, as they intend to be shared. An event is often indicated by a verb in natural language. It therefore seems to be appropriate to make use of the results of linguistic analysis of verbs, specifically of the theory of verb-valency frames. Linguistically based approaches are introduced in section 3. The paper proceeds from John Sowa's thematic roles and the theory of verb-valency frames to propose the general conceptual framework for process ontology which is introduced in section 4.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 83–90, 2021. © Tribun EU 2021

2 Different approaches to event and process specification

With the development of artificial intelligence, it became necessary to depict via conceptualization and ontology the time-dependent and variable phenomena in particular. In a number of contexts and approaches, the concepts of *process* and *event* overlap and these terms are treated as synonyms.[1] However, John Sowa made an essential distinction between them and I am going to proceed from his distinction in this paper. Sowa in [3, p. 220] suggests that "*processes* can be described by their starting and stopping points and by the kind of changes that take place between. [...] In continuous process, which is the normal kind of physical process, incremental changes take place continuously. In a discrete process, which is typical of computer programs or idealized approximations to physical process, changes occur in discrete steps called *events*, which are interleaved with periods of inactivity called *states*."

In order to be able to handle processes, it is important to make some idealization to regard them as discrete processes and divide them into static parts called *states* and into the parts of the change of some state to another state, called *events*. Hence the crucial distinction between the concept of event and process is that the event is some part of the process. Sowa in [3, p. 220] defines process as "an evolving sequence of *states* and events, in which one of the states or events is marked *current* at a context-dependent time called #now."

A similar approach is also applied in the well-known informatics representation, namely the *state-transition diagrams* for discrete processes. They represent states with circles and events by the arrows that connect the circles. Finite-state machines are the most widely used version of state-transition diagrams. The same approach was used also by Carl Adam Petri in [4] when designing his *Petri nets* in 1962. The events are called *transitions* in Petri nets and the states are called *places*.

McCarthy in [5] introduced a representation called *situation calculus* as a logical formalism designed for representing and reasoning about dynamical domains and change. This calculus was later modified by Reiter in [6]. From the logical point of view, situation calculus is a sorted, second-order language with equality. There are three sorts: *situations, actions* and *ordinary objects*, and these sorts can be quantified. A dynamic world is modelled as progressing through a series of *situations*, which are conceptualized as states reachable by some action. Actions are what make the dynamic world change from one situation to another when performed by agents.[2]

Another very important concept in situation calculus is *fluent*. According to situation calculus, fluent is the relation or the function whose last argument is a situation. Fluents are situation-dependent functions used to describe the effects of actions and they are changed by actions that have their preconditions and effects. While actions, situations, and objects are elements of the domain, fluents are modelled as either predicates or functions. Lin in [8, p. 649] presents the following examples of two types of fluents in situation calculus: "There are two kinds of them, *relational* fluents and *functional* fluents. The former has only two values: true or false, while the latter can take a range of values. For instance, one may have a relational fluent called *handempty* which is true in a situation if the robot's hand is not holding anything. We may need a relation like this in a robot domain. One may also have a functional fluent called *battery-level* whose value in a situation is an integer between 0 and 100 denoting the total battery power remaining on one's laptop computer."

---

[1] Bach in [2] called events, states and processes collectively *eventualities*. Barwise and Perry in [3] use the term *situation* in this context.

[2] However, according to the later version of situation calculus developed by Reiter, a situation is a finite sequence of actions, i.e. a period (history) and not a state, see the web source [7].

One may have noticed that there is no autonomous concept for an event (or process) in the situation calculus and it applies the term of action in process specification. According to [8, p. 649], "to describe a dynamic domain in the situation calculus, one has to decide on the set of actions available for the agents to perform, and the set of fluents needed to describe the changes these actions will have on the world." As is the case with situation calculus, *event calculus* also uses the term action to treat events and conceptualize the time-varying properties or fluents. Event calculus was first presented by Kowalski and Sergot in [9] and was further extended by Shanahan and Miller in [10]. Event calculus represents the effects of actions on fluents, the conditions that can change over time. In his comparison of situation and event calculus, Mueller in [11, p. 671] emphasizes that "like situation calculus, event calculus has actions which are called events, and time-varying properties or fluents. In situation calculus, performing an action in a situation gives rise to a successor situation. Situation calculus actions are hypothetical, and time is tree-like. Otherwise, in event calculus, there is a single timeline on which actual events occur."

Hanzal, Svátek and Vacura in [12] provide a general survey of ontologies for modelling events and demonstrate how the dichotomy of *continuants* (entities that persist through time as wholes) and occurrents (entities that are not wholly present at every moment) is incorporated into several well-known foundational ontologies. They survey KR Ontology, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), PURO, and certain other chosen ontologies based on Web Ontological Language (OWL): The Event Ontology, The Simple Event Model Ontology (SEM), Linking Open Descriptions of Events (LODE). They summarize these approaches in the following way: "The surveyed OWL ontologies for modelling events generally share the basic structure, although they differ in certain details: same things are modelled using different 'modelling styles'. What is always central is the class of events whose instances have time properties and are connected to other entities – place, agents etc. – using dedicated properties. In some cases, there are additions to this basic model, for example modelling of different views (SEM)." The authors suggest that classes of different things dispersed in different models are merely subsumed under the common class of events, which gives rise to a relatively flat hierarchy that would be difficult to make sense of as a whole. They propose the following tentative classification of kinds of events into four categories to remedy the problem:

- *C1*, Actions. They assume an explicit or implicit deliberate agent performing them.
- *C2*, Happenings. They cover the situations when "something happened" without being initiated by a deliberate agent.
- *C3*, Planned "social" events. Besides being planned, they typically put emphasis on the spatio-temporal frame rather than on concrete participants.
- *C4*, Structural components of temporal entities. These events are "more arbitrary" than those falling under other categories and can be viewed as "regions", however, as merely temporal (and not spatio-temporal) ones. [12, p. 193]

3 Linguistically based process ontology

Ontological commitments and conceptualization carried out by ontology depend on the goals and purposes of the respective application. When designing an ontology, it is very important to find a balance between the fact that the ontology is designed to achieve the goals of the application and the ability to share such an ontology in the broader context, thus also outside of the interested team that created it. A necessary condition in order for an ontology to be shared is the respect for the role of conceptualized terms in natural language.

Each process can be constituted from the series of events and each event can be specified by a verb in natural language. The semantics of the respective verb is provided via its valency frame. For the linguistic theory of verb-valency frames, see [13]. In general, valency is the ability of a verb (or another word class) to bind other formal units, i.e. words, which cooperate to provide its meaning completely. These units are so-called *functors* or *participants* or *case roles*. Thus, the valency of a verb determines the number of arguments (participants) controlled by a verbal predicate. Valency participants can play an obligatory or a facultative role. One might consider, for example, the verb *chastise*. This verb has two obligatory participants *who* (agent) and *whom* (patient). In addition, this verb can be connected with other facultative participants which express inter alia locality and time such as in the following sentence: *A teacher chastises a student in the school early in the morning.* It would be useful to classify verb participants into types according to their semantics. There are many classifications, however, of the participant types described in the literature, for instance in [13]. Three approaches to classification, according to the two valency dictionaries for the Czech language VALLEX (see [14]) and VerbaLex (see [15]) and John Sowa's approach, are briefly compared in [16].[3] John Sowa also provides his own classification and uses the term *thematic roles* for the verb-valency participants. His summary of all the thematic roles can be found in [3, pp. 506-510]. Here are two examples of formalization of natural language sentence in his conceptual graphs:

---

[3] A very detailed comparison of these three classifications was provided in [17].

Eve bit an apple, conceptualization: [Person: Eve] ←(Agnt) ←[Bite] →(Ptnt) →[Apple]; *Agent* as an active animate entity that voluntarily initiates an action,

Destination as a goal of a spatial process, example: *Bob went to Danbury*: [Person: Bob] ←(Agnt) ←[Go] →(Dest) →[City: Danbury]. For details, see [3, pp. 508-510].

An analysis of sentences with such a complex structure is particularly important when building up a multi-agent system (MAS) with deliberative agents.[4] In general, there is no central dispatcher; the system is driven by messaging so that each autonomous agent though being resource-bounded, can make less or more rational decisions. In addition, by communicating with other fellow agents as well as with their environment, agents are able to learn new concepts and enrich their ontology and knowledge base so that their behaviour is dynamic. Dynamic aspects of agents' reasoning embrace the appropriate conceptualization of participants of activities in their ontology. In the next section a general conceptual framework of process ontology based on the theory of verb-valency frames is proposed.

4 A general proposal for the conceptual framework of process ontology

A similar conceptual framework has been also introduced in [18], namely as a general framework for the logical classification of Wh-questions and possible answers to such questions in a multi agent system. We can distinguish between processes that are based on *actions* of deliberative agents and processes that are based on *passive events* like 'turning pale', 'subsiding', etc., which are not intentional. In [12], these types of processes are classified as C1 (Actions) and C2 (Happenings) in accordance with the above-mentioned classification. A *process* is divided into at least two *states* and one *event*. An event starts the change of state to some other state and is triggered by the respective action of some deliberative agent or some passive event. Hence, actions and passive events are what make the dynamic world change from one state to another. We will call actions and passive events *activities* in general. Each activity can involve other objects that are called its *participants*.

Consider the example of the process of 'going of an agent'. This process is divided into the state$_1$ in which the agent is standing. The action start going changes this state into the state$_2$ in which the agent is going. The measure of the process's granularity depends on the aims of the application that the ontology serves for. For instance, if we want to capture the speed changes, we need to specify the process in more detail. Each speed change has to be captured by adding accelerate and decelerate actions to the ontology.

The starting point of building a process ontology is to distinguish between *static objects* (*static entities*) such as concrete individuals and necessary relations between their properties and *dynamic entities* such as *activities* which are detected by some special types of verbs. The proposed analysis makes use of Tichý's formulation where such verbs are called episodic verbs. Tichý in [19] draws a distinction between episodic and attributive verbs. Episodic verbs (e.g. *drive, tell*, etc.) express the actions of objects or people as opposed to attributive verbs (e.g. *is heavy, looks speedy*) that ascribe some empirical properties to individuals. Both static and dynamic entities are characterised by their further specification. Static entities can be characterised by their properties and attributes, dynamic entities relating to activities can be characterised by the special relationships between activities and their participants.

---

[4] For more details on the multi agent systems in general, see, for instance, [18].

Concerning static entities, from the linguistic point of view, the properties assigned to them are usually denoted by a copular verb + adjective or noun. Typical copular verbs are *is, am, are, ..., appear, seem, look, sound, smell, taste, feel, become* and *get*. In the conceptual analysis of a given domain, it is useful to distinguish between two basic classes of characteristics of static objects. They are relatively stable properties of objects (these characteristics usually remain unchanged over some life-span time) and dynamic empirical facts about these objects. The former can be called 'substantive' properties and the latter 'accidental' properties. For instance, according to the laws of physics and biology, if an individual is born as a person, then during its life-span it cannot become, say, a dog or a vase. Hence, being a person is a substantive property of such an individual. On the other hand, the property of being a student is accidental; one and the same person contingently becomes a student or stops being a student. Other accidental characteristics of the person-type individuals can be, for example, weight, height, age etc. *Substantive properties* are those that individuals have *nomically* necessarily, while *accidental properties* are possessed by individuals purely contingently.

Concerning process ontology, processes are composed of at least one event and two states. States can be formed by some activity (*Petr is standing, Petr is going*), or they are simply the states of affairs (*Apple is red*). On the other hand, events are always triggered by some *activity*. Each activity has an *actor* (who/what is doing the activity) and *participants* of activity. Thematic role or the type of a participant, such as Agent, Patient, Beneficiary, Destination, Instrument, etc., expresses the role that a noun phrase plays with respect to the activity described by a governing verb. The number and the categories of participants depend on the respective domain of interest and the functions of the system of agents. If we want to conceptualize, for example, a 'colour change', we have to include the activity of changing the colour in our conceptualization. It will therefore depend on whether we focus on the agent that causes the colour change, or we will take the colour change as an unintentional change (for example, if it is a natural event). In the first case, the state1 of one of the process may be the situation that the object has some colour. The activity of painting changes this state into state2 in which the object has another colour than in its initial state. The state is specified here by some entity and its attribute 'colour' which is the respective colour. The activity 'to paint' is then specified by the Agent of this activity, the Patient of the activity (the painted object) and by the Manner of activity execution (quickly, in the respective colour, etc.).

5 Conclusion

In this paper, different approaches to process and event ontology have been introduced to obtain basic definitions of the main concepts important for ontology of processes. The proposed approach is based on distinguishing between a static and dynamic part of the domain of interest. This division is based on some necessary idealization and may certainly be reductive. The world is too complex, however, and each effort of conceptualization has to be basically reductive by its very nature. When performing conceptualization, we have to leave out the details which are not fundamental from our point of view and the aims of the intended application.

The proposed conceptual framework follows the usage of the terms in existing ontologies and also their basic meanings in natural language. The specification of processes is based on the concept of activity which is based on Tichý's distinction between episodic and attributive verbs and the theory of verb-valency frames. Process is composed of at least one event and two states, where an event starts the change of state to some other state. Events are triggered by the activities, which can be actions of deliberative agents, or passive events like 'turning pale', 'subsiding', etc. Activities are the dynamic part of the domain. Each activity can concern other objects which are called *participants* according to the theory of verb-valency frames and are modelled as specific relations between the activity and involved objects.

Acknowledgements. The work on this paper was supported by the project *JG_2020_005 Times, events, and logical specification* of Palacký University and Grant of SGS No. SP2021/87.

References

1. Bach, E. On Time, tense and aspect: An essay in English metaphysics. In: R. Bauerle, C. Schwarze and A. von Stechnow (eds.) Meaning Use and Interpretation, pp. 19-38. New York: de Gruyter (1983).
2. Barwise, J., Perry, J. Situations and Attitudes. Cambridge, MA: Bradford Books, MIT Press. (1983).
3. Sowa, J. F. Knowledge representation (logical, philosophical, and computational foundations). Pacific Grove, CA: Brooks Cole Publishing Co (2000).
4. Petri, C. A. Kommunikation mit Automaten. Ph.D. Theses, University of Bonn, Bonn, German. English translation in technical report RADC-TR-65-377, Griffiss Air Force Base (1966).
5. McCarthy, J., Hayes, P. J. Some philosophical problems from the standpoint of artificial intelligence. In: Machine Intelligence, vol. 4, pp. 463–502 (1969).
6. Reiter, R. Knowledge in action: logical foundations for specifying and implementing dynamical systems. Cambridge: The MIT Press (2001).

7. Reiter, R. The situation calculus ontology. Electronic News Journal on Reasoning about Actions and Change (1998). `https://www.ida.liu.se/ext/etai/rac/notes/1997/09/index.html`, last accessed 2021/10/30
8. Lin, F. Situation calculus, In: F. van Harmelen, V. Lifschitz and B. Porter (eds.) Handbook of Knowledge Representation, pp. 649–669. Elsevier B.V. (2008).
9. Kowalski, R., Sergot, M. A logic-based calculus of events. In: New Generation Computing 4 (1), pp. 67–95 (1986).
10. Miller, R., Shanahan, M. Some alternative formulations of the event calculus, In: A. C. Kakas, F. Sadri (eds.) Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski Part II, Lecture Notes in Computer Science, pp. 452–490. Berlin, Heidelberg: Springer (2002).
11. Mueller, E. Event Calculus. In: F. van Harmelen, V. Lifschitz and B. Porter (eds.) Handbook of Knowledge Representation, pp. 671-708, Elsevier (2008).
12. Hanzal, T., Svátek, V., Vacura, M. Event categories on the semantic web and their relationship/object distinction. In R. Ferrario and W. Kuhn (eds.), Formal Ontology in Information Systems (pp. 183-196). Amsterdam: IOS Press (2016).
13. Horák, A. Verb Valency and Semantic Classification of Verbs. In Proceedings of TSD'98, pp. 61-66, Brno (CR): Masaryk University (1998).
14. Lopatková, M., Žabokrtský, Z., Kettnerová, V. VALLEX 2.5. – Logical structure of the lexicon (2006). `http://ufal.mff.cuni.cz/vallex/2.5/doc/structure_en.html#sec:frame`, last accessed 2021/10/30
15. Hlaváčková, D., Horák, A. VerbaLex - New Comprehensive Lexicon of Verb Valencies for Czech. In: Computer Treatment of Slavic and East European Languages, pp. 107-115 (2006).
16. Číhalová, M. Event ontology specification based on the theory of valency frames. In: T. Welzer, H. Jaakkola, B. Thalheim, Y. Kiyoki, N. Yoshida (eds.) Frontiers in Artificial Intelligence and Applications, Information Modelling and Knowledge Bases XXVII, pp. 299-313, Amsterdam: IOS Press (2016).
17. Číhalová, M. Jazyky pro tvorbu ontologií (Languages for ontology building). Ph.D. Thesis, VŠB-Technical University of Ostrava, Ostrava, The Czech Republic (2011).
18. Číhalová, M., Duží, M. Modelling dynamic behaviour of agents in a multi-agent world; logical analysis of Wh-questions and answers. Submitted to the Logic Journal of the IGPL.
19. Tichý, P. The semantics of episodic verbs. In: Theoretical Linguistics, vol. 7, pp. 263-296 (1980).
20. Wooldridge, M. An introduction to multi-agent systems. London: John Wiley & Sons (2009).

Towards Domain Robustness of Neural Language Models

Michal Štefánik and Petr Sojka

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`stefanik.m@mail.muni.cz`, `sojka@fi.muni.cz`

Abstract. This work summarises recent progress in generalization evaluation and training of deep neural networks, categorized in data-centric and model-centric overviews. Grounded in the results of the referenced work, we propose three future directions towards reaching higher robustness of language models to an unknown domain or its adaptation to an existing domain of interest. In the example propositions that practically complement each of the directions, we introduce novel ideas of **a**) dynamic objective selection, **b**) language modeling respecting the token similarities to the ground truth and **c**) a framework of additive component of the loss utilizing the well-performing generalization measures.

Keywords: Generalization · Debiasing · Domain extrapolation · Domain adaptation · Domain robustness · Neural language models

> "Education is the most powerful weapon we can use to change the world."
> Nelson Mandela

1 Introduction

Deep language models have found their application in a wide variety of tasks, ranging among other aspects in their semantic complexity and a domain of applicability. While a domain of some applications can be bound, commonly, we can not afford to utilize a specialized model for every possible *domain*, i.e., a set of samples of which we apply the language model, conditioned by a distinct situational and pragmatic background. Furthermore, our domains of interest might not even be preliminary known, as is often the native case in generative tasks, such as neural machine translation, summarization, or paraphrasing; think, for example, of a variety of domains for which a general-purpose machine translation system can be applied.

The exceeded reliance of the models on characteristics of a single training domain shows as an increasing problem only with an increased expressivity of the deep architectures, which are for the first time able to accurately model the non-representative relations not easily apparent to their maintainer. As one of the first, McCoy [24] demonstrates a reliance of state-of-the-art transformer

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 91–103, 2021. © Tribun EU 2021

model on heuristical shortcuts on language inference [42], specifically on a lexical and subsequence overlap between the premise and hypothesis. Belinkov [2] and Berard [4] show fragility of neural machine translation models to typos and misspelling, and vocabulary shift, respectively, both common for non-canonical domains that the systems are usually not trained on. A large branch of work follows, either in aims to empirically identify domain-specific biases in commonly-used data sets [39,14,29,16], or in aims to heuristically eliminate these biases in data [24,27,48].

This paper brings an introductory overview of the limited set of existing methods that address the qualitative discrepancy of applying the model to samples of different domain(s), regardless of the specific type of domain shift between the training and target domain.

Section 2, overviews the existing methods based on resampling the training domain samples or exposing the domain shift by using the data from two different domains. Further, in Section 3, we extend this list for a domain that *adjust* the standard training process via adjusting the objective of the training process.

Finally, in Section 4 we outline the open ends implied by the results of the preceding studies, which could lead to an enhancement of the model's domain robustness. We aim to describe these common directions tangibly enough to be utilizable in future research. We thoroughly describe a single technical proposition for each of the three outlined directions and leave its empirical evaluation to the subsequent studies.

2 Extrapolation using Data

Data approaches aim to utilize the available samples, possibly categorized by their domain of origin, in order to minimize test loss on samples of the domain of interest. In the scope of a well-recognized branch of work labeled as *domain adaptation*, the training situation is denoted by the availability of source domain X_s, which can be interpreted as a random variable generating the samples x_s with their corresponding labels y_s. Further we denote a target domain X_t, i.e. a domain of application, with a limited amount of $(x_t, y_t) \in X_t$, where it holds that $|X_t| < |X_s|$, or in some situations, where the amount of $y_t \in X_t$ is limited.

In the more extreme case referred in the literature as an evaluation for *domain generalization*, we restrict the training process to access only samples of source domain(s) X_s, and the samples of X_t are a priori unavailable. Arguably, this situation better corresponds to open-domain applications such as open machine translation.

2.1 Impact of Data Subsampling

Data Selection approaches aim to resample the samples used for the training process in order to maximize the generalization ability of the eventual model.

Denoising strategies elaborate on a hypothesis that some samples are less representative to the task of interest than others. Among more straightforward approaches, Lin [22] picks the "clean" set of samples according to their perplexity to the linear base model, keeping in the training set only the ones with low perplexity. Later, Moore [25] seeks to pick the samples x_s that minimally affect the sum log-likelihood of the model updated according to x_s. Similarly, Yarowsky [45] pick the training subsample based on a threshold on the sample output *confidence*. Zhou [49] iteratively applies the same strategy using an ensemble of three estimators, only picking the top-n most-confident samples, possibly avoiding the mangle confidence calibration, and refers to this approach to as *tri-training*.

An interesting, yet more complex approach, referred to as *Product-of-Experts* is introduced by [15]. Here, an ensemble of relatively small classifiers is used to *debias* the training samples by computing a dot product of class-wise *logits* of the ensemble and possibly discarding the samples for which the ensemble disagrees the most. Sanh [33] applies this approach to the training transformers model and finds interesting performance gains on out-of-domain performance. Similarly, Utama [39] identify the possibly-biased samples as the ones reaching high confidence only for a single one of the ensembled models and consecutively *weights* the training samples by their chance of exposing bias. In the broader scope, these approaches fit well into the PAC-Bayesian framework [40], roughly stating that if for the selected model M empirical error bound ϵ_M, then for the error for an *ensemble* E of such models it holds that $\epsilon_E \leq \epsilon_M$.

2.2 Ability to Distinguish Domains

Another approach to domain generalization leads through an *exposition* of the domain discrepancies, which is a necessary precondition for the model to comprehend and possibly to model it. This is theoretically supported by the work of Locatello [23], concluding that *distributional robustness* is not possible without the exposition of both *data* and *model* inductive biases. Bengio [3] demonstrates how these biases can be utilized by the model to fit the causal structure of the data and evaluate this ability in the situation where the data-specific inductive biases are known.

There are simpler ways how domain discrepancies can be effectively communicated to the model. For example, Shah [34] minimizes the Wasserstein distance of internal model representations between the samples of source and target domain, X_s and X_t. Jiang [17] first trains the domain classifier C_d distinguishing domains X_s and X_t and subsequently *weights* the samples $x_s \in X_s$ in the training by their correspondence to X_t as given by the confidence of C_d. Chadha [5] enhances out-of-domain performance of adapted model by adding so-called *maximum mean discrepancy loss* to the training objective, given by $\max(dist(x_s, x_t)) : x_s \in X_s, x_t \in X_t$.

3 Extrapolation and Training Process

The adjustments to the training process have proved to increase the distributional robustness of the final model in different variations. We identify that the authors of empirically-successful works in generalization use the regularization element, which corresponds to a specific well-performing generalization measure. Hence, we first describe popular evaluation measures and then describe the specific adjustments of the training process leading to a model with better generalization.

3.1 Evaluation of Extrapolation

In a large-scale study on image classification, Jiang [18] shows that the measures of so-called *spectral graph complexity* [28], *sharpness* of the parametrized space [19], or PAC-Bayesian measures [40], similar to the introduced Product-of-Experts, correlate the highest to the empirical out-of-domain performance of the convolutional model. Later, Dziugaite [11] dispute some of these results, reproducing the experiments in enhanced, fine-grained methodology, showing that the high average correlations of some measures, such as the spectral complexity, systematically fail under specific domain shifts.

Perhaps surprisingly, these studies agree upon the low correlations of the standard regularization techniques such as dropout or norms regularization, suggesting that an application of techniques sufficient to avoid in-domain overfitting might not be sufficient for reaching distributional robustness.

3.2 Training Process Adjustments

A large branch of studies shows that regularizing the training process using the referenced generalization measures positively impacts the distributional robustness of the model. However, note that most of the following studies were applied in evaluating image classification, with questionable relevance to transfer learning settings.

Barlett [1] uses spectral complexity as a norm in the training process of the AlexNet convolutional network and theoretically demonstrates that this property corresponds to the network generalization ability. Similarly, Foret [12] uses sharpness as an additive term of loss, computed on locally-surrounding inputs as an additive component of the training loss. In addition to increasing out-of-domain accuracy, the resulting model demonstrates higher robustness to noisy training in-domain samples. Referring to the process as “debiasing”, Utama [39] utilize the commonly-evaluated PAC Bayesian confidence estimate in predictions in loss weighting.

Other adjustments give some insights into the impact of the composition of transfer learning objectives. While Teney [37] or Wang [26] demonstrate the cases where adaptation to a single domain harms out-of-distribution robustness of the model, Wu [43] concludes that adapting to multiple data sets can enhance the end model generalization. Additionally, Tu [38] reporting a positive impact of multitask learning to model's out-of-distribution accuracy, or by Xie [44] for additive consistency regularization in the training objective.

4 Future Perspectives

Grounded in the referenced studies and results, we now describe three potential directions that could mitigate the exposition of inductive biases in the language models and, consequently, reach their higher generalization ability. We enrich each one of these directions with a practical proposition that contributes to the described direction.

Overall, we observe that the strategy of interaction with a model during the training has a significant impact on the model's generalization ability, just like the teacher's methods and interaction have a principal effect on the student's performance. All of the introduced directions elaborate on interaction strategies towards the model on training time.

4.1 Impact of Objectives Curricula

"If we examine ourselves, we see that our faculties grow in such a manner that what goes before paves the way for what comes after." J. A. Comenius [8]

While many of the mentioned studies, for example, [5,43,38] enrich the training objective with an exposition of the domain discrepancies and their respective biases with reported positive impact to generalization, it is not clear how the specific strategies of doing so vary in effectiveness and efficiency. For instance, Gururangan [13] concludes that it is always beneficial to perform a fine-tuning to a domain or a task of interest by sequentially applying the different objectives, Tu [38] apply a concurrent objective schedule. Additionally, as some objectives might be easier than others, it is likely that some objectives overweight others over time, mitigating the further convergence, possibly necessary for learning the corner cases [38].

We propose to systematically enhance our comprehension of the performance of models in the different objectives: do we somewhat loose grasp of a general language understanding, reflected, for example, in Masked or Causal language modeling accuracy [10,31], or Denoising [21], when fine-tuning for a token or sequence classification on end task? If this degradation is significant, as suggested, for example, by the results of Popel [30], it motivates the results for a more complicated schedule of an application of objectives.

> *If a fine-tuning on end objective degrade performance of other relevant objectives, we are motivated to utilize a non-sequential schedule of these objectives in the common adaptations.*

We propose to confront a standard sequential schedule of the optimization of the objective with the novel ones. We aim to investigate at least the two strategies outlined in Figure 1: a "striped" schedule strategy, where the loss of *all* objectives

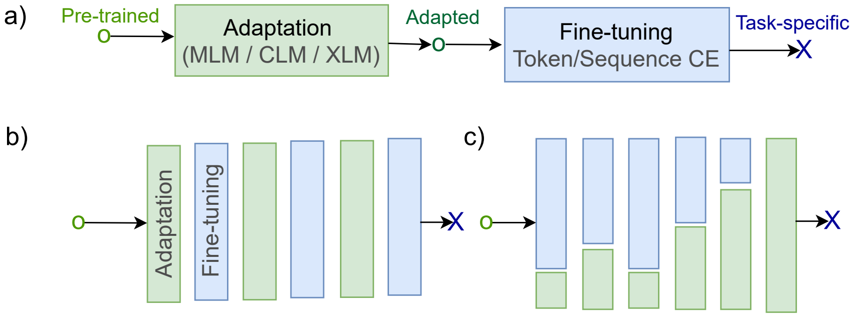


Fig. 1: Illustrative comparison of basic objective sampling strategies. Traditionally, domain adaptation is performed in sequential strategy (**a**). Presumably, a combined sampling strategy (**b**), could avoid performance decay of the unscheduled, yet relevant objective(s), as reported for instance by Popel [30]. A dynamic sampling (**c**), based for example on a state of the validation loss, could further eliminate this performance decay.

is included in each training step, and a candidate of the groups of "dynamic" strategies, where the objective selection is determined by a heuristic based on the immediate loss of given objective.

4.2 Softer Objectives

> "The proper education of the young does not consist in stuffing their heads with a mass of words, sentences, and ideas dragged together out of various authors, but in opening up their understanding to the outer world, so that a living stream may flow from their own minds, just as leaves, flowers, and fruit spring from the bud on a tree." J. A. Comenius [8]

The continuous over-parametrization of deep language models brings qualitative gains even by following the same, well-established objectives on the same, limited amount of training resources of end tasks, as shown for instance by [10,9]. Still, it makes sense to ask whether the commonly-used objectives expose the characteristics of the learned task in an *efficient* manner, both with respect to the computational resources and often expensive supervised data resources.

Consider the cases of Masked, or Causal language modeling, where 15% of randomly-selected tokens is masked. Presuming the Zipf law holding for the natural language artifacts in all its levels (from morphology to semantic of, e.g., coreference or entity recognition), the chance of exploiting the long tail of less common artifacts remains long underrepresented. On the other hand, an exposition of the trivial artifacts, e.g., a resolution of the correct pronoun, when the referenced subject is already referenced in the unmasked segment, occurs commonly.

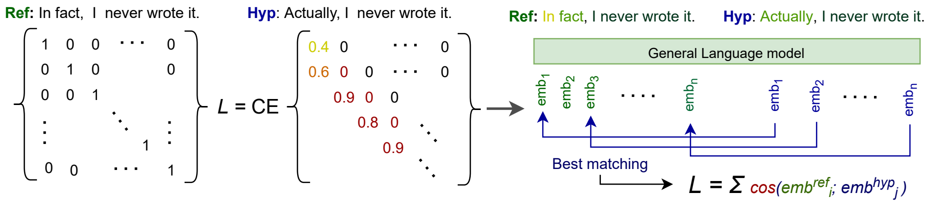

Ref: In fact, I never wrote it.
Hyp: Actually, I never wrote it.
L = CE
0.4
0.6
0.9
0.8
0.9
Ref: In fact, I never wrote it.
Hyp: Actually, I never wrote it.
General Language model
emb1
emb2
emb3
embn
emb1
emb2
embn
Best matching
L = Σ cos(emb^ref_i; emb^hyp_j)

Fig. 2: Instead of using the cross-entropy (CE) exact-matching objectives, we propose to elaborate into using "soft" objectives, able to distinguish between the different levels of *inexact* matching. As an example, we propose to compute a loss of sequence-to-sequence training objective *irrespective* of the relative ordering of the tokens in the reference and hypothesis. Similar to the evaluation of Zhang [46], the objective would first find the best-possible matching between two sets of tokens based on the token embeddings, and only then computes the value of the loss as a sum of minimal possible distances of every token in the hypothesis. Note that such objective is still differentiable on a sequence level.

> *We should ask whether the commonly-used objectives expose the full variety of the learned task in an* efficient *manner, as the efficiency will always be a qualitative bottleneck for many low-resource or domain-specific applications.*

The inefficiency, as well as the potential of objectives improvement, is exploited by the approach of ELECTRA model [7]. ELECTRA uses a simpler language model to exchange some words in the pre-training corpora. The language model is trained to distinguish the synthetically-exchanged tokens in the token classification objective instead of using the classic MLM objective. Using this approach, authors report 30x speedup of convergence while reaching very similar performance on a set of GLUE [41] tasks.

Another significant work in this direction is the one of Szegedy et al. [36], which introduces commonly-used Label smoothing nowadays. In this training strategy, the "true" distribution of labels to which the model's loss is computed is not discrete, i.e., in the form of a one-hot vector of a size of several classes $|C|$. Instead, it has a form of a vector with the values of $\frac{\epsilon}{|C|}$ on the positions of non-expected category, and a value of $1 - \epsilon$ on the true-category position, where ϵ remains a free parameter, usually set in $\langle 0.05; 2 \rangle$. Such smoothing of the objective is shown to minimize in-domain test error [36] and can improve model generalization ability [6].

These results motivate us to revisit the commonly-used objectives, where a speed of convergence and generalization can be defining factors of model's end quality, for instance, in a neural machine translation of under-resourced languages or non-canonical domains [32].

We follow with a brief motivational introduction to the problem and a proposition of one specific machine translation objective following the call for softer objectives. The approach is also summarised in Figure 2.

The standard neural machine translation objective is to minimise the cross-entropy (CE) loss between an *expected* pseudo-probabilistic distribution over model's vocabulary, for each token P^E_i given by the model, and a *true* token $y'_i \in Y^T$ given by a set of *reference translations*. P^E is *conditioned* by both the tokens of the source sequence $x_{1\dots n}$ and the *previous* tokens $y_{1\dots n-1}$. The cross-entropy token-level loss $\mathcal{L}$ is then defined as:

$$\mathcal{L}(X, y'_{1\dots n}) = \sum_{i=1}^{|n|} CE(P^E_i(X, y'_{1\dots i-1}), y'_i)\,.$$

Utilising $\mathcal{L}$ in the training process, the model is trained to *predict* all P^E_i any unknown X, but compared to the training, on inference, P^E_i is conditioned by the *previously-predicted* tokens $y_{1\dots i-1}$ instead of the tokens of the reference $y'_{1\dots i-1}$.

Among other aspects, $\mathcal{L}$ implies that if the model generates one extra token or omits one token at the beginning of generation, all the subsequently-generated tokens will be sanctioned the same as if the model generated the remaining output randomly. A similar penalization is backpropagated if the model fully *paraphrases* the reference. Such a loss origin might arguably cause the model to *overfit* the syntax of the training domain, or might be the reason why the other objectives, such as Denoising [21] significantly enhance a *fluency* of output, as compared to the described Causal language modeling, as in GPT [31].

One of the simple approaches to eliminate this problem is to start with picking a reference token y'_j which is *best-matching* to the evaluated x_i. A separate, discriminative language model can provide the representations of the matched tokens, similarly to [7]. The pairwise distance of the tokens can be estimated using the max-product approach as proposed in BERTScore [47], using the many-to-many matching utilizing Wasserstein distance [20], or using any other differentiable token-level distance measure.

4.3 Objectives Utilizing Generalization Measures

> "What we demand is vigilance and attention on the part of the master
> and the pupils." J. A. Comenius [8]

A relatively specific direction towards higher robustness of language models is outlined by the works utilizing the approximations of measures that correlate well with empirical out-of-distribution performance. These works overviews Section 3.2. Even though some of the incorporated measures do not consistently correlate to out-of-distribution performance, from a limited number of the referenced applications, it seems that the model is always able to utilize the adjacent information efficiently.

Task-specific training objectives can be extended with an additive component, in a form outlined in Equation (1).

$$\mathcal{L}(M) = (1-\alpha)\mathcal{L}_{Obj}(M) + \alpha\mathcal{L}_{Meas}(M) \tag{1}$$

> *To enhance model's distributional robustness, a task-specific training objective $\mathcal{L}_{Obj}$ can be additively complemented with a differentiable instance of the generalization measure $\mathcal{L}_{Meas}$.*

The measures that highly correlate with out-of-distribution accuracy of the model can be utilized to effectively regularize the final objective $\mathcal{L}$ favouring the property associated with distributional robustness. We overview some of such generalization measures in Section 3.1.

We identify two challenges in training objective design. The first one is in designing a differentiable and computationally-feasible approximation of the generalization measure. Foret [12] demonstrates that the valuation of sharpness of the parametrized space requires a valuation for all the inputs of the parametrized, application-dependent distance. It is not clear if a similar representative valuation would be feasible in the NLP domain.

The second challenge lies in designing the evaluation measures well-correlated with out-of-distribution performance and their representative evaluation. For example, Dziugaite [11] shows that the measures that correlate highly in one context might correlate poorly under different shifts. A representative evaluation of the generalization ability of the measure requires identification of all valid biases, which is not feasible, implying that the evaluation of generalization measures will remain merely the point estimates of unknown shift.

We can still escape this uncertainty in designing the generalization measures reflecting the features of the problem, which we intuitively consider to be invariant to the data domain or problem on hand. Such features could, for instance, reflect the shared linguistic properties of the natural language.

5 Conclusion

This work outlines the three directions of addressing the unwanted data biases of language models, which is an extensively reported problem inherently raised from the expressivity of the deep models.

We aim to motivate the research in these three directions, providing a shared framework and referencing the current work showing initial, promising results.

We acknowledge that there might be multiple unforeseen obstacles in any proposed directions that will only identify in practice. We argue that any contribution towards more robust language models has immediate implications for most of the applications in the NLP field. Many of the commonly-used solutions already rely on transformers and can even be seen to expose unknown, notorious biases, as shown, e.g., in [35]. At the same time, a limited extrapolation ability of the models remains a blocker for applying modern NLP in more niche domains, where little annotated data is available due to the size or audience background.

References

1. Bartlett, P.L., Foster, D.J., Telgarsky, M.: Spectrally-Normalized Margin Bounds for Neural Networks. In: Proc. of the 31st International Conference on Neural Infor-

mation Processing Systems. pp. 6241–6250. NIPS '17, Curran Associates Inc., USA (2017). `https://doi.org/10.5555/3295222.3295372`

2. Belinkov, Y., Bisk, Y.: Synthetic and Natural Noise Both Break Neural Machine Translation. CoRR **abs/1711.02173v2** (2018), `https://arxiv.org/abs/1711.02173v2`
3. Bengio, Y., Deleu, T., Rahaman, N., Ke, N.R., Lachapelle, S., Bilaniuk, O., Goyal, A., Pal, C.: A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms. In: Proc. of International Conference on Learning Representations (2020), `https://openreview.net/forum?id=ryxWIgBFPS`
4. Berard, A., Calapodescu, I., Dymetman, M., Roux, C., Meunier, J.L., Nikoulina, V.: Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In: Proc. of the 3rd Workshop on Neural Generation and Translation. pp. 168–176. ACL, Hong Kong (Nov 2019). `https://doi.org/10.18653/v1/D19-5617`
5. Chadha, A., Andreopoulos, Y.: Improving Adversarial Discriminative Domain Adaptation. CoRR **abs/1809.03625v3** (2018), `https://arxiv.org/abs/1809.03625v3`
6. Chen, B., Ziyin, L., Wang, Z., Liang, P.P.: An Investigation of how Label Smoothing Affects Generalization. CoRR **abs/2010.12648** (2020), `https://arxiv.org/abs/2010.12648v1`
7. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. CoRR **abs/2003.10555v1** (2020), `https://arxiv.org/abs/2003.10555v1`
8. Comenius, J.A.: Didáctica magna. Amsterdam (1649), `https://webspace.ship.edu/cgboer/comenius.html`, The Great Didactic, translated by M. W. Keatinge 1896
9. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised Cross-lingual Representation Learning at Scale. CoRR **abs/1911.02116v2** (2020), `https://arxiv.org/abs/1911.02116v2`
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805v2** (2018), `https://arXiv.org/abs/1810.04805v2`
11. Dziugaite, G.K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., Roy, D.M.: In Search of Robust Measures of Generalization. CoRR **abs/2010.11924v2** (2021), `https://arxiv.org/abs/2010.11924v2`
12. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-Aware Minimization for Efficiently Improving Generalization. CoRR **abs/2010.01412v1** (2021), `https://arxiv.org/abs/2010.01412v1`
13. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: Proc. of the 58th Annual Meeting of the ACL. pp. 8342–8360. ACL (Jul 2020), `https://aclanthology.org/2020.acl-main.740.pdf`
14. Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S.R., Smith, N.A.: Annotation Artifacts in Natural Language Inference Data. In: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 107–112. ACL, New Orleans, Louisiana (Jun 2018). `https://doi.org/10.18653/v1/N18-2017`
15. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation **14**(8), 1771–1800 (Aug 2002). `https://doi.org/10.1162/089976602760128018`
16. Iyer, S., Dandekar, N., Csernai, K.: First Quora Dataset Release: Question Pairs (2017), `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`

17. Jiang, J., Zhai, C.: Instance Weighting for Domain Adaptation in NLP. In: Proc. of the 45th Annual Meeting of the ACL. pp. 264–271. ACL, Prague, Czech Republic (Jun 2007), `https://aclanthology.org/P07-1034`
18. Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic Generalization Measures and Where to Find Them. CoRR **abs/1912.02178v1** (2020), `https://arxiv.org/abs/1912.02178v1`
19. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. CoRR **abs/1609.04836v1** (2017), `https://arxiv.org/abs/1609.04836v1`
20. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From Word Embeddings To Document Distances. In: Bach, F., Blei, D. (eds.) Proc. of International Conference on Machine Learning. vol. 37, pp. 957–966. PMLR, Lille, France (Jul 2015), `http://proceedings.mlr.press/v37/kusnerb15.html`
21. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proc. of the 58th Annual Meeting of the ACL. pp. 7871–7880 (2020), `https://aclanthology.org/2020.acl-main.703.pdf`
22. Lin, S., Tsai, C., Chien, L., Chen, K., Lee, L.: Chinese language model adaptation based on document classification and multiple domain-specific language models. In: Kokkinakis, G., Fakotakis, N., Dermatas, E. (eds.) Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997. ISCA, Rhodes, Greece (Sep 1997), `http://www.isca-speech.org/archive/eurospeech_1997/e97_1463.html`
23. Locatello, F., Bauer, S., Lucic, M., Gelly, S., Schölkopf, B., Bachem, O.: Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. CoRR **1811.12359v4** (2019), `https://arXiv.org/abs/1811.12359v4`
24. McCoy, T., Pavlick, E., Linzen, T.: Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In: Proc. of the 57th Annual Meeting of the ACL. pp. 3428–3448. ACL, Florence, Italy (Jul 2019). `https://doi.org/10.18653/v1/P19-1334`
25. Moore, R.C., Lewis, W.: Intelligent Selection of Language Model Training Data. In: Proc. of the ACL Conference. pp. 220–224. ACL, Uppsala, Sweden (Jul 2010), `https://aclanthology.org/P10-2041`
26. Nie, Y., Wang, Y., Bansal, M.: Analyzing Compositionality-Sensitivity of NLI Models. CoRR **abs/1811.07033v1** (2019), `https://arxiv.org/abs/1811.07033v1`
27. Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., Kiela, D.: Adversarial NLI: A new benchmark for natural language understanding. In: Proc. of the 58th Annual Meeting of the ACL. pp. 4885–4901. ACL (Jul 2020). `https://doi.org/10.18653/v1/2020.acl-main.441`
28. Pitas, K., Davies, M.E., Vandergheynst, P.: PAC-Bayesian Margin Bounds for Convolutional Neural Networks. CoRR **abs/1801.00171v2** (2018), `https://arxiv.org/abs/1801.00171v2`
29. Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., Van Durme, B.: Hypothesis Only Baselines in Natural Language Inference. In: Proc. of the Seventh Joint Conference on Lexical and Computational Semantics. pp. 180–191. ACL, New Orleans, USA (Jun 2018). `https://doi.org/10.18653/v1/S18-2023`, `https://aclanthology.org/S18-2023`
30. Popel, M., Tomková, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar, O., Žabokrtský, Z.: Transforming machine translation: a deep learning system reaches news transla-

tion quality comparable to human professionals. Nature Communications **11**(4381) (2020). `https://doi.org/10.1038/s41467-020-18073-9`

31. Radford, A., Narasimhan, K.: Improving Language Understanding by Generative Pre-Training (2018), `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`
32. Ramponi, A., Plank, B.: Neural unsupervised domain adaptation in NLP—A survey. In: Proc. of the 28th International Conference on Computational Linguistics. pp. 6838–6855. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). `https://doi.org/10.18653/v1/2020.coling-main.603`
33. Sanh, V., Wolf, T., Belinkov, Y., Rush, A.M.: Learning from others' mistakes: Avoiding dataset biases without modeling them. CoRR **abs/2012.01300v1** (2021), `https://arxiv.org/abs/2012.01300v1`
34. Shah, D.J., Lei, T., Moschitti, A., Romeo, S., Nakov, P.: Adversarial Domain Adaptation for Duplicate Question Detection. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proc. of the 2018 Conference EMNLP. pp. 1056–1063. ACL (2018). `https://doi.org/10.18653/v1/d18-1131`
35. Štefánik, M., Novotný, V., Sojka, P.: RegEMT: Regressive Ensemble for Machine Translation Quality Evaluation. In: Proc. of the Sixth Conference on Machine Translation (WMT). pp. 1046–1053. ACL (Nov 2021), `https://www.statmt.org/wmt21/pdf/2021.wmt-1.112.pdf`, poster also available: `https://mir.fi.muni.cz/posters/emnlp-2021-regemt.pdf`
36. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: IEEE Conf. CVPR. pp. 2818–2826. IEEE, Los Alamitos, USA (Jun 2016). `https://doi.org/10.1109/CVPR.2016.308`
37. Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline. In: Proc. of the 57th Annual Meeting of the ACL. pp. 4593–4601. ACL, Florence, Italy (Jul 2019). `https://aclanthology.org/P19-1452`
38. Tu, L., Lalwani, G., Gella, S., He, H.: An Empirical Study on Robustness to Spurious Correlations using Pre-trained Language Models. Transactions of the ACL **8**, 621–633 (Oct 2020). `https://doi.org/10.1162/tacl_a_00335`
39. Utama, P.A., Moosavi, N.S., Gurevych, I.: Towards Debiasing NLU Models from Unknown Biases. In: Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7597–7610. ACL, Online (Nov 2020). `https://doi.org/10.18653/v1/2020.emnlp-main.613`
40. Valiant, L.G.: A Theory of the Learnable. In: Proc. of the Sixteenth Annual ACM Symposium on Theory of Computing. pp. 436–445. STOC '84, ACM, New York, USA (1984), `https://doi.org/10.1145/800057.808710`
41. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: Proc. of the 2018 EMNLP Workshop BlackboxNLP. pp. 353–355. ACL, Brussels, Belgium (Nov 2018), `https://aclanthology.org/W18-5446`
42. Williams, A., Nangia, N., Bowman, S.: A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In: Proc. of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1 (Long Papers). pp. 1112–1122. ACL (2018), `https://aclweb.org/anthology/N18-1101`
43. Wu, M., Moosavi, N., Rücklé, A., Gurevych, I.: Improving QA Generalization by Concurrent Modeling of Multiple Biases. CoRR (2020), `https://arxiv.org/abs/2010.03338v1`
44. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised Data Augmentation. CoRR **abs/1904.12848v1** (2019), `https://arXiv.org/abs/1904.12848v1`

45. Yarowsky, D.: Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: 33rd Annual Meeting of the ACL. pp. 189–196. ACL, Cambridge, Massachusetts, USA (Jun 1995). `https://aclanthology.org/P95-1026`
46. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with BERT. CoRR **abs/1904.09675v3** (2019), `https://arxiv.org/abs/1904.09675v3`
47. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. In: Proc. of International Conference on Learning Representations (2020), `https://openreview.net/forum?id=SkeHuCVFDr`
48. Zhang, Y., Baldridge, J., He, L.: PAWS: Paraphrase Adversaries from Word Scrambling. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proc. of the 2019 Conf. NAACL-HLT. pp. 1298–1308. ACL, Minneapolis, USA (Jun 2019). `https://doi.org/10.18653/v1/n19-1131`
49. Zhou, Z.H., Li, M.: Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering **17**(11), 1529–1541 (2005). `https://doi.org/10.1109/TKDE.2005.186`

Part III

Morphology and Syntax

Approaching Punctuation Errors in the New Proofreader of Czech

Vojtěch Mrkývka

Faculty of Arts, Masaryk University
Arne Nováka 1, 602 00 Brno, Czech Republic
`mrkyvka@phil.muni.cz`

Abstract. As the progress of a new online proofreader of Czech continues, so does the development of particular proofreading modules that make it whole. The position of the punctuation one is rather specific as its inner workings differ from the usual structure. This paper focuses on the design of the punctuation module, its specifics and obstacles which followed or still follow its development process.

Keywords: Proofreading · Punctuation · Regular expressions

1 Introduction

The new online proofreader of Czech is a new tool developing at the Faculty of Arts, Masaryk University since 2018. Contrary to similar products, this project aims to (hopefully) address a broader spectrum of errors, not ending with a spellchecker but starting with it. Using knowledge of both Czech language and computational linguistics gained at PLIN[1], the team aims to create a rule-based system by formalising existing basic research results supplemented by own findings. This paper will focus on one specific part of the tool – the punctuation module, its specific nature within the proofreader and obstacles that were or yet have to be overcome.

2 About the proofreader

Although the nature of the proofreader varied in time[2], the current (and hopefully final) solution – Plinkorektor[3] – has a form of singular API with a modular internal structure communicating with the user interface to present results (see Fig. 1). However, the final goal for the API is to be on any specific user interface fully independent.

As mentioned above, the API consists of multiple internal modules called simultaneously as soon as their requirements are fulfilled (see Fig. 6). Additionally, the current version allows the user to specify whether he or she wants to

---

[1] Computational linguistics study programme at Faculty of Arts, Masaryk University

[2] For more information, see my previous papers on the topic[1,2,3].

[3] `https://korektor.plin.cz/`

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 107–113, 2021. © Tribun EU 2021

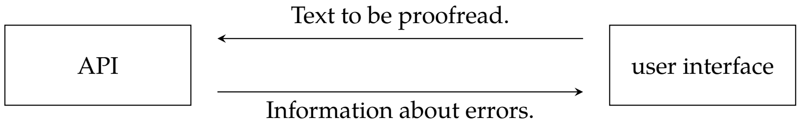


Fig. 1: Communication between API and the user interface.

call only a specific part of the module portfolio, omitting dependencies that are redundant for the selection.

The team working on the API creates the detection rules for the different types of user errors (spelling, commas, or subject-verb/subject-object grammatical agreement) that need to be provided with the correction overlays. Rules, which are the outcome of these overlays, are strongly tied to prior tokenisation operating solely on replacement operation.

3 The punctuation module

The punctuation module is based on the bachelor thesis of Zbyněk Michálek [4]. It contained a detailed list of regular expressions (44 in total), which can be used for automatic detection and correction of selected issues. These expressions were implemented in the user interface, automatically correcting some of the errors before calling the API. From the current point of view, this solution was unfortunate because it added (weak) API dependency on the user interface, preventing the Plinkorektor API from being fully used in a different environment. The only logical solution was to migrate these rules to the API.

Most of the proofreading modules natively work with the tokens using shallow parsing grammars for the SET analyser [5] to detect and mark the problematic sequences. However, this is not the case with the punctuation module. As mentioned above, its determination is based on regular expressions, so it is working with text independently of the tokens; however, the API needs the token mapping for the correct output production. Fortunately, the match object from Python's `re` package can provide information about on which character position the regular expression match starts, ends or both using `start()`, `end()` or `span()` methods respectively. Using a pointer array, the context of the matches could be determined easily[4] (see Fig. 2). However, the `re` package later showed to be insufficient, as it does not operate with POSIX classes. Fortunately, the alternative `regex` package can be used in its place. The immodest goal of the author is for `regex` to replace `re` in the future as it provides more functions (for example, already mentioned POSIX class compatibility) while maintaining maximum backwards compatibility with `re`[6]. Sadly, even the package replacement did not fully fulfil all the needs. Although the base of regular expressions is usually the same across the programming languages, further nuances can make the specific expression unusable within different environments. For example, for detection of space followed by a comma[5] Michálek uses the expression `[:blank:],`; however, in Python, POSIX classes have to be encapsulated in another pair of brackets as `[[:blank:]]`, in this case.

---

[4] Additional context limitation in case of some of the rules was to include additional groups into the expressions themselves for start/end/span methods to operate with.

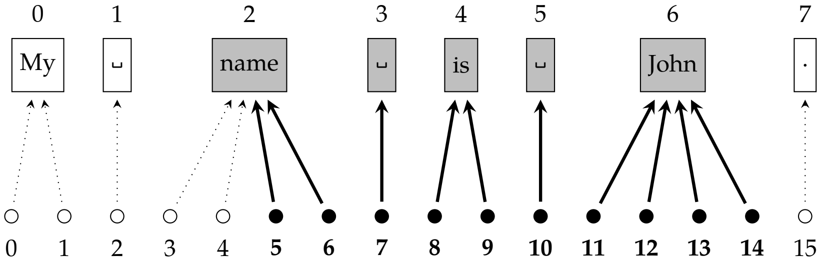


Fig. 2: An example of mapping the regular expression (`[me.*n]` in this case) on tokens.

The correction rules for these expressions can be divided into two approaches, one using regular expressions only for error detection and the other for both detection and replacement. Using the example mentioned above, the space token can be selected using capturing group (`([[:blank:]]),`) and removed by the simple *replace with nothing* rule (see Fig. 3).

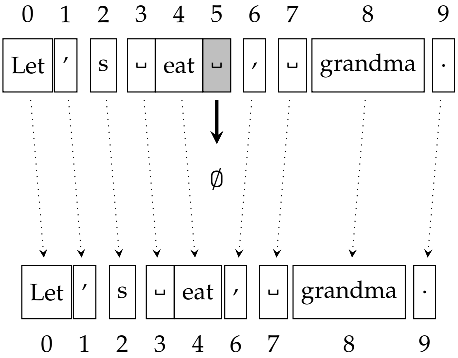


Fig. 3: An example of the *replace with nothing* rule. The space before comma in the sentence “Let’s eat , grandma.” is replaced with an empty string virtually removing the space as a result.

It should be mentioned that Michálek provided replacement patterns for all of his regular expressions; however, as the original intended usage was different[6] he uses capturing groups (if he uses them at all) for the parts of the expression that shall be kept after correction (e.g. to be used in replacement pattern) rather than the parts to replace. The second approach is based on using these rules ignoring the token structure of corrections by moving the result of the replacement pattern into the first affected token, leaving the others blank[7]. However, this approach is discouraged due to problematic compatibility with other modules, as some of the expressions can span over many tokens (see Fig. 4).

---

[5] In Czech, there should never occur space before a comma. In the position after a comma, the space is usually present, but there is an exception for numeric expressions.

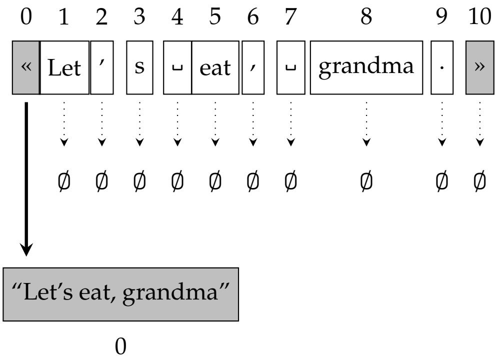


Fig. 4: An example of replacing the whole quotation segment (because of incorrect quotation marks) with single token when using regular expression replacement rules.

As on a related problem can be looked at the Michálek’s expressions themselves. As mentioned, their intent was to be used strictly as automatic correction and do not always fulfil Plinkorektor needs. For example, the expression `\u00A7([:blank:]?)([0-9])` used to replace space after § for no-break space cannot be used as is, as no-break space is part of `[:blank:]` POSIX class and it would create the false-positive message. Aside from this, opinion-based issues need to be resolved when dealing with automatic corrections, but in other cases can be left for the user to decide. For example, Michálek uses the expression `\?\?+` to replace all cases of multiple question marks with exactly three[8]. However, in the case of Plinkorektor, the user can select if he or she wants to use one or three question marks when exactly two were input. Similarly, there is a question of whether the expression to remove additional whitespace before the colon and the one missing after it should be treated as one issue or two separate ones. This can relate to the abovementioned dilemma whether use regular expressions also for the replacement purposes, as splitting of selected expressions (or using suitable capture groups) can help keep the tokens intact (compare Fig. 4 with Fig. 5).

---

[6] Michálek intended to use his rules solely for automatic correction of given issues, however, the philosophy of Plinkorektor is to provide users information about which corrections can be used as automatic but leaving the choice on them.

[7] The text is retokenised every time the API is called, so the change of token structure will not affect further API calls.

[8] He works similarly with exclamation marks.

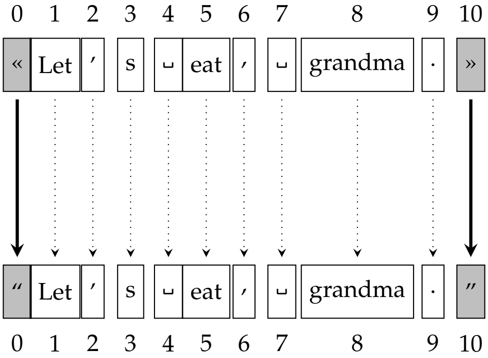

0 1 2 3 4 5 6 7 8 9 10
« Let ’ s ␣ eat , ␣ grandma . »
“ Let ’ s ␣ eat , ␣ grandma . ”
0 1 2 3 4 5 6 7 8 9 10

Fig. 5: An example of replacing the quotation segment by parts (because of incorrect quotation marks) with the most of the tokens left in place.

4 Common issues with the API

Lastly, there are issues with the API itself that prevent the punctuation module from being better as part of it instead of a separate tool (for example, as part of the user interface as it is right now). The main problem is that the current API is still relatively slow to be entirely usable in the production environment (see Fig. 6). Although there are still options to speed specific parts up, some modules will always be slower than others. The supplementary option is to give users better ways to call parts of the API independently to, for example, check the text with the fast modules first with additional correction by the slower modules after they finish their processing.

5 Conclusion

The new online proofreader of Czech still has many issues that need to be addressed, and the ongoing development of the punctuation module (currently at circa 10%) is no exception. The situation presented above and the whole of the Plinkorektor issues can be summarised as quantity over difficulty situation, meaning there is a minimal number of problems, which can be considered hard. However, easy ones come in such quantity that progress is not always optimal. On the other hand, looking at the overall work done versus to be done, the production-wise usable product is undoubtedly just around the corner.

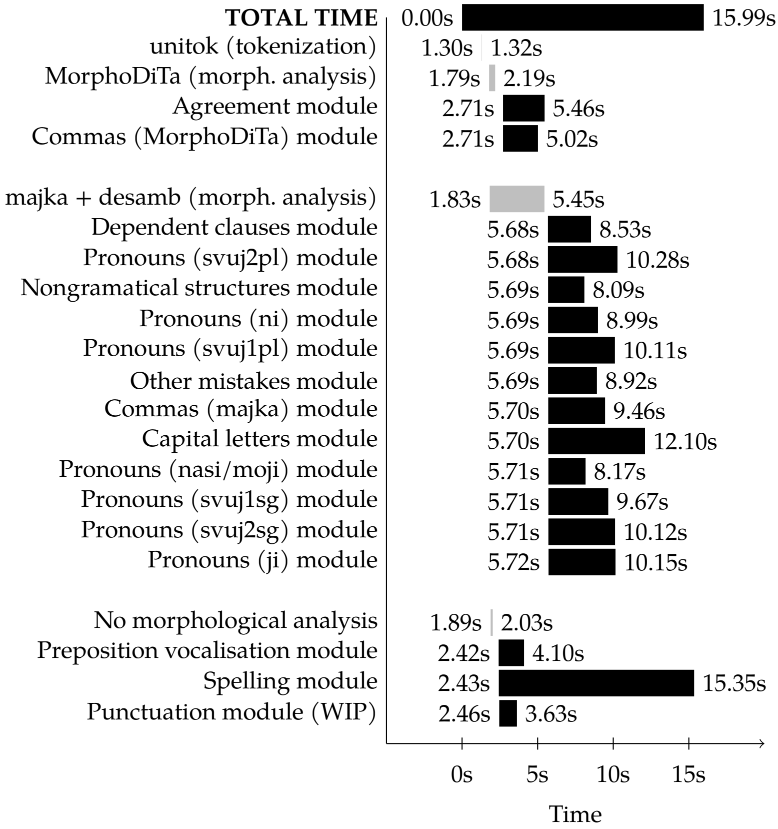


Fig. 6: Runtime of different modules in the API.

Acknowledgements. The work was supported by the project of specific research *Gramatika a lexikon češtiny* (Grammar and lexicon of Czech; project no. MUNI/A/1181/2020).

References

1. Mrkývka, V.: Webové rozhraní pro automatický jazykový korektor češtiny [online]. Diplomová práce, Masarykova univerzita, Filozofická fakulta, Brno (2018 [2021-10-28])
2. Mrkývka, V.: Towards the New Czech Grammar-checker. In Horák, A., Rychlý, P., Rambousek, A., eds.: Proceedings of the Twelfth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2018, Brno, Masaryk University (2018) 3–8
3. Mrkývka, V.: Recent Advancements of the New Online Proofreader of Czech. In Horák, A., Rychlý, P., Rambousek, A., eds.: Proceedings of the Thirteenth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2019, Brno, Masaryk University (2019) 43–47

4. Michálek, Z.: Algoritmizace hromadných oprav vybraných typograficko-pravopisných jevů českého jazyka [online]. Bakalářská práce, Masarykova univerzita, Filozofická fakulta, Brno (2016 [2021-10-28])
5. Kovář, V., Horák, A., Jakubíček, M.: Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech. In: Human Language Technology. Challenges for Computer Science and Linguistics, Berlin/Heidelberg, Springer (2011) 161–171
6. Barnett, M.: regex · pypi [online] (2021 [2021-10-28])

Evaluating the State-of-the-Art Sentence Alignment System on Literary Texts

Edoardo Signoroni

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`e.signoroni@mail.muni.cz`

Abstract. Sentence alignment is a useful task with many applications in Natural Language Processing and Digital Humanities. This paper presents an evaluation of Vecalign, the state-of-the-art method for automatic sentence alignment, on two bilingual corpora built from literary texts. This preliminary study shows that Vecalign performs well for literary texts and gives insights on its remaining issues through a qualitative evaluation of the output alignments.

Keywords: Parallel corpora · Automatic alignment · Literary text

Introduction

Sentence alignment is the Natural Language Processing (NLP) task of taking parallel documents split into sentences and finding a bipartite graph which matches minimal groups of sentences that are translation of each other [20]. In other words, to find target sentences with the same meaning to that of the source segments in multilingual texts [19].

This task is important to build bilingual corpora on which statistical Machine Translation (MT) systems could be trained. While neural MT approaches seem to be performing much better with sizable amounts of data, Kim et al. (2020) [6] shows that supervised and semi-supervised baselines outperform the best unsupervised systems.

Good alignment is also crucial for lexicography, as it can be leveraged to display parallel concordances and to find translation equivalents, and for terminology extraction.

Parallel corpora alignment is also being used in Digital Humanities (DH) with various purposes, such as historical language learning [10] or version alignment for medieval texts [8].

After a brief overview of the related work (Section 1), and a description of the methodology employed for this work (Section 2), the paper evaluates the performance of Vecalign [20] through a qualitative manual analysis (Section 3) of its automatic alignment of two corpora built from literary texts.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 115–124, 2021. © Tribun EU 2021

1 Related Work

This section will present some related work relevant to this study, firstly describing currently employed sentence alignment methods, and then briefly covering their application on literary texts and in DH.

1.1 Sentence Alignment Methods

The first automatic alignment methods were simple: they align sentences according to their length in words [4] or characters [5]. These algorithms do not work for text with sentences that have the same length, such as list of names or dates. Other systems worked with correspondence rules [17].

Newer approaches employ either external dictionaries or by training a translation model on the parallel text itself [22,9]. They also add some heuristics, such as limiting the search space to be near the diagonal. These systems, however, do not work with small texts because the occurrences of a given word are few. More recent methods introduced MT-based scoring [15,16], such as BLEU [11].

Steingrimsson et al. (2020) [19] review the current literature on the topic of sentence alignment and parallel corpora filtering. They then devise a new pipeline for aligning and filtering parallel corpora in sparse data conditions building on existing methods, such as those in Sennrich et. al. (2011) [16] and Artetxe et al. (2018a) [1]. Their proposed method is language pair independent and assumes unaligned bitexts and monolingual corpora.

The state-of-the-art systems use bilingual sentence embeddings, with their similarity used as the scoring function for alignment [20]. This is the method that it is employed for this paper, and it will be further described in Section 2.1.

The latest work on sentence alignment was presented at the Fifth Conference on Machine Translation (WMT2020), which featured a shared task on "Parallel Corpus Filtering and Alignment for Low-resource conditions" [7].

1.2 Work in Digital Humanities

Steinbach and Rehbein (2019) [18] demonstrate a pipeline for the parallelization and the annotation transfer for literary texts. For the sentence alignment they use Bleualign [16].

Meinecke, Wrisley, and Jänicke (2019) [8] employ the gensim implementation [14] of fastText [3] word embeddings and sentence embeddings similarity to compare and align different versions of the same medieval text.

The use of automatic alignment in DH is varied and broad. Some examples include Pataridze and Kindt (2018) [12], the Rosetta Stone project[1], or Zhekova et al. (2015) [24]. It seems common for these works to present their own domain-specific tools, such as UGARIT [2]. It is out of the scope of this paper to survey all the different application of automatic alignment systems in DH, nonetheless the examples above should give and idea of the variety of uses it has.

---

[1] `https://rosetta-stone.dh.uni-leipzig.de/rs/home/`

[2] `http://ugarit.ialigner.com/index.php`

None of the work on literary texts or in DH seems to take advantage of Vecalign [20] as the state-of-the-art alignment system.

2 Methods

This section will discuss the methodology of this work, presenting the tools and the corpus on which they were tested.

2.1 Vecalign and LASER

Vecaling[3] was chosen as the automatic alignment system for two main reasons: i. it is the current state-of-the-art system; ii. it seems to be still untested on literary texts.

Vecaling propose a new scoring function based on the similarity of bilingual sentence embeddings. The method computes sentence embedding similarity scores with cosine similarity normalized with randomly selected embeddings. It then averages adjacent pairs of sentence embeddings in both documents and align these approximate embeddings, iteratively refining this alignment using the original embeddings and a small window around them.

Following the Vecalign paper, LASER[4] was used to compute the sentence embeddings. This tool is based on the architecture for language agnostic sentence embeddings presented in Artexte and Schwenk (2019) [2].

2.2 Corpora

Two corpora were used for the experiments: i. a manually aligned version of Lewis Carrol's "Alice's Adventures in Wonderland"[5]; ii. three versions of J.R.R. Tolkien's "The Hobbit".

The first corpus consists of 823 sentences from "Alice's Adventures in Worderland" manually aligned and reviewed by András Farkas in nine languages. Only English and Italian were considered. This corpus was considered as a possible gold-standard to automatically evaluate the performance of Vecalign, however this was proven to be problematic for several reasons, which will be mentioned in the following section.

The second corpus is from J.R.R. Tolkien's book "The Hobbit" [21]. Three unaligned editions in three different languages (English, Czech, and Italian) where collected. The full .txt files averaged around 2.200 lines.

Table 1 summarizes the size of the two corpora.

---

[3] `https://github.com/thompsonb/vecalign`

[4] `https://github.com/facebookresearch/LASER`

[5] Retrieved from `https://farkastranslations.com/books/Carroll_Lewis-Alice_in_wonderland-en-hu-es-it-pt-fr-de-eo-fi.html`

	number of lines	**number of sentences**
alice_en	824	824
alice_it	824	824
hobbit_en	1989	5770
hobbit_it	2372	5342

Table 1: Number of lines in the .txt and number of sentences after preprocessing.

3 Experiments and Evaluation

Since we are not dealing with text scraped from the web, or processed with Optical Character Recognition (OCR) algorithms, or otherwise overly noisy data, not much preprocessing was needed.

Alice's corpus did not need specific preprocessing: it was easily downloaded in .csv form and the sentences for English and Italian were stored in separate .txt files. LASER sentence embeddings were trained with standard parameters and Vecaling was run with default settings. The output alignment was stored as a .csv file. Since Vecaling gives its output alignments as pairs of lists of sentences IDs, these were leveraged to add the text of the sentences to the .csv to qualitatively evaluate the resulting alignment. In case of alignments between multiple sentences, these were split by the special character *$* in order for them to be distinguishable in the .csv.

The Hobbit's corpus underwent some preprocessing stages. The text was first obtained in .doc format, it was then converted into .txt to be processed. By doing so, some features of the book, such as illustrations, images, and page numbers were lost. The text was split into sentences with [13], even if LASER is capable of handling training of sentence embeddings from raw text. Future work may address if this step actually has any impact on the output, since a preliminary observation has shown that the text was divided in a different number of sentences by LASER and Stanza. Sentences were stored in a separate .txt file for each language. LASER and Vecalign were again used with their default configuration. The resulting alignments were stored in three .csv files.

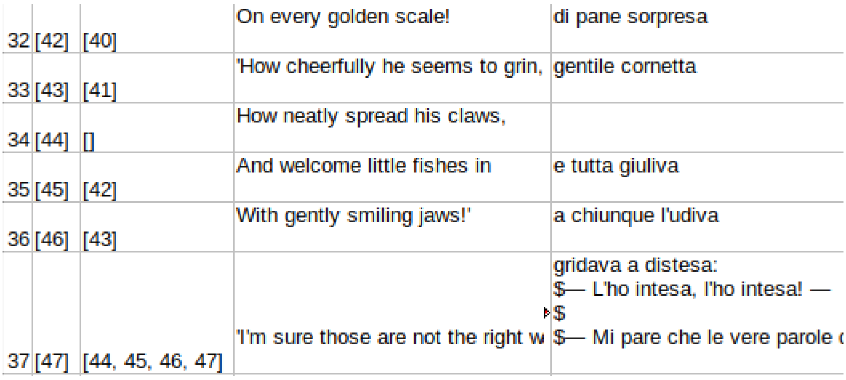
	start	**mid**	**end**	**average**
alice_en_it	85	98	100	**94,33**
hobbit_en_it	83	96	99	**92,67**

Table 2: Scores for the manual evaluation batches: the first (start), central (mid), and last (end) one hundred EN to IT alignments and the overall average score for each corpus.

Evaluation proved to be more complex than anticipated. Several automated methods were considered to evaluate the alignment quality. The Alice corpus was considered as reference for the design phase of the evaluation, since it has a gold standard. After taking an overview of the resulting outputs, an automated method of evaluation was tentatively devised. However, all of the proposed methodologies proved to be flawed. For example, a simple automated comparison between proposed alignment and gold standard alignment was revealed to be ineffective since it did not consider 1-to-many and many-to-one alignments. A MT-based method based on word lists comparison and BLEU score was considered, but proved to be unwieldy. Devising an automated evaluation method for The Hobbit corpus was even more challenging, since there was non gold standard available.

It was then decided to provide a qualitative evaluation of the results by manually assigning a score (0 for a bad alignment and 1 for a good alignment) to three batches of 100 alignments, one from the beginning, one from the main body, and one from the end. The scores were then averaged. Albeit simple, this method still provided some useful insights on the performance and the issues of Vecalign. The scores are given in Table 2

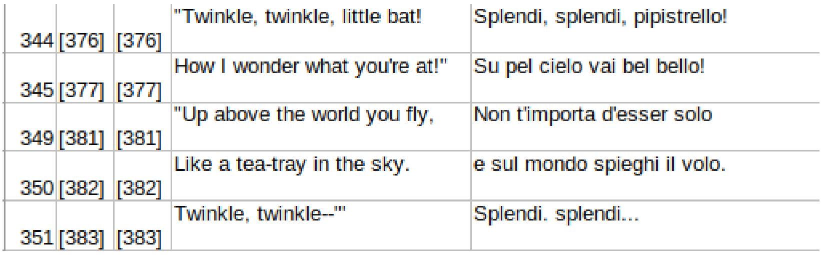
On the Alice corpus, 94.33% out of the 300 evaluated alignments where judged to be good. The first batch was the worst one, with 85/100, while the other two had respectively 98/100 and 100/100. Some interesting facts were uncovered by the analysis.



32	[42]	[40]	On every golden scale!	di pane sorpresa
33	[43]	[41]	'How cheerfully he seems to grin,	gentile cornetta
34	[44]	[]	How neatly spread his claws,	
35	[45]	[42]	And welcome little fishes in	e tutta giuliva
36	[46]	[43]	With gently smiling jaws!'	a chiunque l'udiva
37	[47]	[44, 45, 46, 47]	'I'm sure those are not the right w	gridava a distesa: $— L'ho intesa, l'ho intesa! — $ $— Mi pare che le vere parole

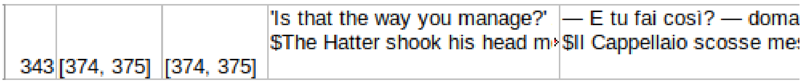
Fig. 1: The adaptation of a popular rime that confounds the alignment. The Italian version is not the translation of the English text.

First, while many of the alignments (a.) were correct, often they were not exact translations of the source sentences. This seems to hold true for the whole text, but some peculiar cases are rimes such as in a. 31 through 37 (Fig. 1) where not translated at all, but adapted to reflect the target culture. This also holds true for other translation choices as well, such as in a. 43 where the original reference to William the Conqueror is changed to Napoleon. The different adaptation seems to be irrelevant with regards to the performance if it is limited to a single



344	[376]	[376]	"Twinkle, twinkle, little bat!	Splendi, splendi, pipistrello!
345	[377]	[377]	How I wonder what you're at!"	Su pel cielo vai bel bello!
349	[381]	[381]	"Up above the world you fly,	Non t'importa d'esser solo
350	[382]	[382]	Like a tea-tray in the sky.	e sul mondo spieghi il volo.
351	[383]	[383]	Twinkle, twinkle--"'	Splendi. splendi...

Fig. 2: Another localized popular rime. In this case, however, the alignment is maintained.

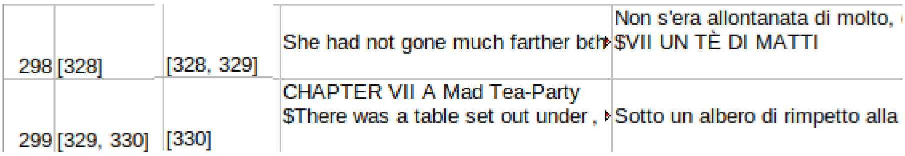


343	[374, 375]	[374, 375]	'Is that the way you manage?' $The Hatter shook his head m	— E tu fai così? — doma $Il Cappellaio scosse me

Fig. 3: A 2-to-2 alignment due to direct discourse markers and punctuation.

word, but in case of longer segments, it can lead to misalignment, such as in the aforementioned a. 31-37. The algorithm reports higher alignment cost for sections such as these.

Second, there is a general tendency to generate a 2-to-2 alignment between a short phrase with direct dialogue and a longer following sentence. This is most probably due to the presence of punctuation. However, this does not impact the alignment quality since the sentences are correctly paired (Fig. 3)



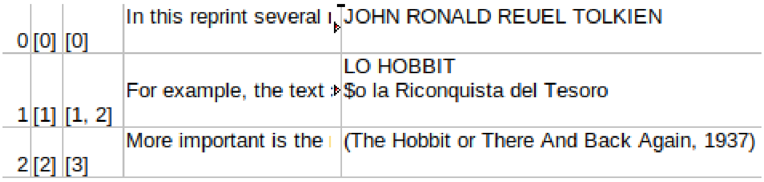
298	[328]	[328, 329]	She had not gone much farther bef	Non s'era allontanata di molto, $VII UN TÈ DI MATTI
299	[329, 330]	[330]	CHAPTER VII A Mad Tea-Party $There was a table set out under ,	Sotto un albero di rimpetto alla

Fig. 4: A misaligned chapter heading.

Third, often the chapter header is misaligned in a 1-to-2 or a 2-to-1 alignment together with the preceding or the following sentence (Fig. 4). Different choices in the typesetting of, for example, the direct discourse marker, did not impact the performance of the algorithm.

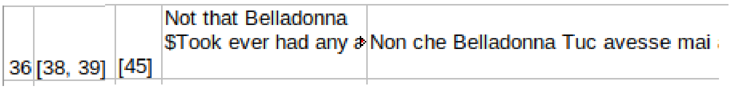
On the Hobbit corpus, 92.67% out of the 300 evaluated alignments were judged as correct. Again the first batch was the worst one, with 83/100, while the others scored 96/100 and 99/100. This corpus was slightly noisier than the Alice one, since the two Hobbit books differed in some editorial choices.

The first 10 alignments are all incorrect: the beginning of the book is completely different in the two editions, nonetheless Vecalign paired sentences in a miscellaneous assortment of 1-to-1, 1-to-2, and 1-to-many alignments



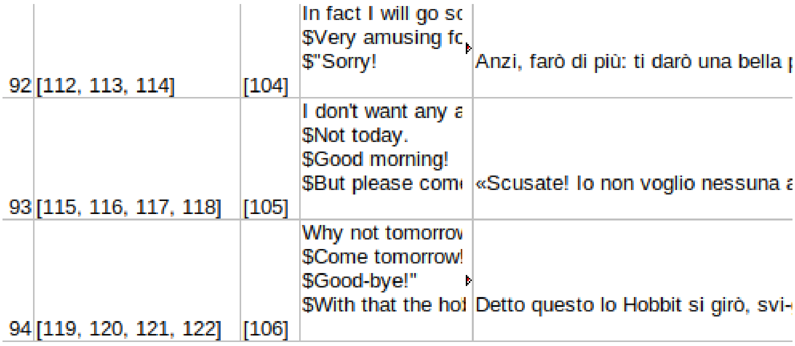
0	[0]	[0]	In this reprint several	JOHN RONALD REUEL TOLKIEN
1	[1]	[1, 2]	For example, the text	LO HOBBIT $o la Riconquista del Tesoro
2	[2]	[3]	More important is the	(The Hobbit or There And Back Again, 1937)

Fig. 5: A section of the misaligned beginning of the Hobbit corpus.



36	[38, 39]	[45]	Not that Belladonna $Took ever had any a	Non che Belladonna Tuc avesse mai

Fig. 6: A split named entity: "Belladonna Took".



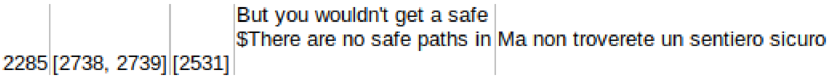
92	[112, 113, 114]	[104]	In fact I will go sc $Very amusing fc $"Sorry!	Anzi, farò di più: ti darò una bella
93	[115, 116, 117, 118]	[105]	I don't want any a $Not today. $Good morning! $But please com	«Scusate! Io non voglio nessuna a
94	[119, 120, 121, 122]	[106]	Why not tomorrov $Come tomorrow! $Good-bye!" $With that the hol	Detto questo lo Hobbit si girò, svi-

Fig. 7: An erroneous many-to-1 alignment. Only the last one is correctly aligned.

(Fig. 5) The ideal output should have been a series of blanks on both sides, alternatively.

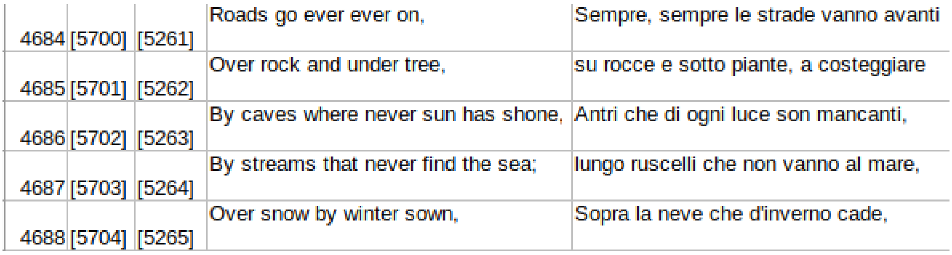
In some cases, e.g. a. 32, 36, and 37, preprocessing tricked the algorithm into creating a 2-to-1 alignment. For example, an unrecognized named entity could be split in the middle, generating a new sentence (Fig. 6). These preprocessing problems are likewise found in other sections of the text, e.g. a. 79-80, giving rise to unwanted many-to-many alignments(Fig. 7).

These problems, however, seem to be more due to differences in the tokenization model between the two languages, than due to Vecalign. Nonetheless, they are somewhat useful to this analysis, since they show that Vecalign is not totally impervious to errors when dealing with short sentences, such as in a. 92-94. In other cases, e.g. a. 4730 and 4724, the system coped well with differences in punctuation and sentence structure that influenced tokenization and sentence splitting. Moreover, the Italian version of the text contained some line break markings ("-") inside words, but this seems not to have influenced the quality of the alignment.



2285	[2738, 2739]	[2531]	But you wouldn't get a safe $There are no safe paths in	Ma non troverete un sentiero sicuro

Fig. 8: A missing blank in the target alignment. The second sentence is not in the Italian version.



4684	[5700]	[5261]	Roads go ever ever on,	Sempre, sempre le strade vanno avanti
4685	[5701]	[5262]	Over rock and under tree,	su rocce e sotto piante, a costeggiare
4686	[5702]	[5263]	By caves where never sun has shone,	Antri che di ogni luce son mancanti,
4687	[5703]	[5264]	By streams that never find the sea;	lungo ruscelli che non vanno al mare,
4688	[5704]	[5265]	Over snow by winter sown,	Sopra la neve che d'inverno cade,

Fig. 9: A poem-like section. Most of it is correctly aligned.

Sometimes, a blank was expected, but Vecalign choose to merge the unaligned sentence with the following one. This is the case with a. 2285 (Fig. 8).

Lastly, in the Hobbit as well are found some songs that could be considered rimes or poems, both in structure and content. The a. 4684-4626 are a good example of this case: apart from the last two lines that confound the algorithm, the other are correctly aligned, unlike the first Alice rime. This could be due to the fact that in the Hobbit the poem is translated, and not adapted (Fig. 9).

4 Conclusion and Future Work

This paper described two experiments that tested and evaluated Vecaling, the state-of-the-art method for automatic sentence alignment, on two corpora of literary texts. The system was shown to perform well, even if some issues, such as not optimal handling of blank-aligned sentences and the management of short phrases and sentence boundaries, remain to be resolved.

Future work may address issues in automatic sentence alignment such as dealing with noisy or OCRed text and evaluate the impact of preprocessing, such as sentence splitting and text cleaning, on the final alignment task. Moreover, a good automatic quantitative evaluation framework should be devised to complement qualitative manual evaluation.

English-Czech and Czech-Italian alignments of the Hobbit corpus were computed, but not evaluated, and are available for future research.

Acknowledgements. The author's work is funded by Masaryk University and by Lexical Computing. The author would like to thank his supervisor doc. Mgr. Pavel Rychlý, Ph.D. for the feedback and his colleagues in the NLP lab and at Lexical Computing for the help with practical issues.

References

1. Artetxe, M., Labaka, G., Agirre, E.: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 789–798. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1073, `https://aclanthology.org/P18-1073`
2. Artetxe, M., Schwenk, H.: Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. CoRR **abs/1812.10464** (2018), `http://arxiv.org/abs/1812.10464`
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Brown, P.F., Lai, J.C., Mercer, R.L.: Aligning sentences in parallel corpora. In: Proceedings of the 29th Annual Meeting on Association for Computational Linguistics. p. 169–176. ACL '91, Association for Computational Linguistics, USA (1991). https://doi.org/10.3115/981344.981366, `https://doi.org/10.3115/981344.981366`
5. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. Computational Linguistics **19**(1), 75–102 (1993), `https://aclanthology.org/J93-1004`
6. Kim, Y., Graça, M., Ney, H.: When and why is unsupervised neural machine translation useless? (2020)
7. Koehn, P., Chaudhary, V., El-Kishky, A., Goyal, N., Chen, P.J., GuzmÃ¡n, F.: Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In: Proceedings of the Fifth Conference on Machine Translation. pp. 726–742. Association for Computational Linguistics, Online (November 2020), `https://www.aclweb.org/anthology/2020.wmt-1.78`
8. Meinecke, C., Wrisley, D.J., Jänicke, S.: Automated alignment of medieval text version based on word embeddings (2020). https://doi.org/https://doi.org/10.31219/osf.io/tah3y
9. Moore, R.C.: Fast and accurate sentence alignment of bilingual corpora. In: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas: Technical Papers. pp. 135–144. Springer, Tiburon, USA (Oct 8-12 2002), `https://link.springer.com/chapter/10.1007/3-540-45820-4_14`
10. Palladino, C., Foradi, M., Yousef, T.: Translation alignment for historical language learning. Digital Humanities Quarterly **15**(3) (2021)
11. Papineni, K., Roukos, S., Ward, T., jing Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 311–318 (2002)
12. Pataridze, T., Kindt, B.: Text Alignment in Ancient Greek and Georgian: A Case-Study on the First Homily of Gregory of Nazianzus. Journal of Data Mining and Digital Humanities **Special Issue on Computer-Aided Processing of Intertextuality in Ancient Languages** (Jan 2018), `https://hal.archives-ouvertes.fr/hal-01294591`
13. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020), `https://nlp.stanford.edu/pubs/qi2020stanza.pdf`

14. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), `http://is.muni.cz/publication/884893/en`
15. Sennrich, R., Volk, M.: Mt-based sentence alignment for ocr-generated parallel texts. In: The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010) (11 2010). https://doi.org/10.5167/uzh-38464
16. Sennrich, R., Volk, M.: Iterative, mt-based sentence alignment of parallel texts. In: NODALIDA (2011)
17. Simard, M., Foster, G.F., Isabelle, P.: Using cognates to align sentences in bilingual corpora. In: Proceedings of the Fourth Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages. Montréal, Canada (Jun 25-27 1992), `https://aclanthology.org/1992.tmi-1.7`
18. Steinbach, U., Rehbein, I.: Automatic alignment and annotation projection for literary texts. In: Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. pp. 35–45. Association for Computational Linguistics, Minneapolis, USA (Jun 2019). https://doi.org/10.18653/v1/W19-2505, `https://aclanthology.org/W19-2505`
19. Steingrímsson, S., Loftsson, H., Way, A.: Effectively aligning and filtering parallel corpora under sparse data conditions. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 182–190. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-srw.25, `https://aclanthology.org/2020.acl-srw.25`
20. Thompson, B., Koehn, P.: Vecalign: Improved sentence alignment in linear time and space. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 1342–1348. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1136, `https://www.aclweb.org/anthology/D19-1136`
21. Tolkien, J.R.R.: The Hobbit, or There and Back Again. George Allen & Unwin (1937)
22. Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., Trón, V.: Parallel corpora for medium density languages. Recent Advances in Natural Language Processing IV pp. 247–258 (01 2007). https://doi.org/10.1075/cilt.292.32var
23. Xu, Y., Max, A., Yvon, F.: Sentence alignment for literary texts: The state-of-the-art and beyond. In: Linguistic Issues in Language Technology, Volume 12, 2015 - Literature Lifts up Computational Linguistics. CSLI Publications (Oct 2015), `https://aclanthology.org/2015.lilt-12.6`
24. Zhekova, D., Zangenfeind, R., Mikhaylova, A., Nikolaienko, T.: Sentence-alignment and application of russian-german multi-target parallel corpora for linguistic analysis and literary studies. MATLIT: Materialities of Literature **4**(1), 45–61 (Feb 2015). https://doi.org/10.14195/2182-8830_4-1_3, `https://impactum-journals.uc.pt/matlit/article/view/2182-8830_4-1_3`

Building a Dataset for Detection of Verb Coordinations with a Shared Argument

Helena Medková

Faculty of Arts, Masaryk University Brno,
Czech Republic
`gerzova@phil.muni.cz`

Abstract. Coordinate structures represent a specific linguistic problem relating to questions of sentence boundaries and multiple sentence element [1]. A particular difficulty lies in processing at the level of automatic syntactic analysis of the sentence. To deal with the outlined issue, we decided to use the machine learning classification method, for which it is necessary to prepare a sufficiently large amount of data. This paper presents the methods and procedures we used to build a dataset focused on the phenomenon of verb coordinations that may share an argument in context.

Keywords: Coordination · Zeugma · Syntactic analysis · UDPipe 2 · Brat annotation tool · VerbaLex

1 Introduction

The phenomenon of coordinate structures is a challenging task in natural language processing as it can be a complex problem also for a human annotator.

Difficulties can arise because of the parts of sentence ellipsis, which makes such constructions semantically ambiguous and complete reconstruction of the meaning or the author's intention is not always entirely possible. We show the example of multiple interpretation possibilities on the sentence (1) from corpus czTenTen17 [2]:

> (1) ***Obřad*** *má* ***zachránit*** *a* ***přinést dušim*** *posvátný* ***klid****.* (*The ceremony have to save and bring sacred peace to the soul.*)

In the Czech sentence, we cannot reliably determine if the *ceremony* is the subject that grammatically agrees to the verb (*mít / have to*) or if it is the object of the verb *save*. The coordination could also be ungrammatical if we read the indirect object in the dative (*duším; souls*) as an argument of both coordinated verbs. In practice, such structures tend to be excluded from automatic processing because of their difficulty to handle. [3]

In this paper, we present the dataset building process and a description of the methods we used. The dataset focuses on two predicate coordinations that

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 125–133, 2021. © Tribun EU 2021

share at least one argument in the context of the sentence. As an ungrammatical equivalent of such structures, we consider a zeugma.

An annotated dataset will allow us to use supervised machine learning methods and train a classifier to recognize verb coordinations with a shared argument. Furthermore, it will be possible to compare the benefits of a different approach (than rule-based) to the problem.

2 The coordinate structures

The typical example is the coordination of two verbs that bind the same object (2) [2]. The sentence elements that would be repeated in two sentences are thus brought into the same syntactic position by their deletion from the surface structure in one of the coordinated sentences [4]. These structures allow the writer to avoid the redundancy of words in the sentence when syntactic rules are fulfilled.

> (2) *Tím **zmírňuje** a **odstraňuje pískání** a **hučení** v uších.* (*It reduces and eliminates whistling and tinnitus.*)

We can also find formally equivalent structures in sentences in which the two predicates do not share anything (3) [2].

> (3) *Jde o léky*[...]*, které alergické **příznaky zmírňují** a **brání zhoršení** nemoci.* (*The medicines*[...] *that relieve allergic symptoms and prevent the disease from worsening.*)

The non-grammatical alternative to the structures above is binding two expressions by a single dependent element, where the syntactic rules are not met. The expressed syntactic dependency of the constituent contradicts the required syntactic dependency demanded by one of the conjuncts [5]. See sentence example(4) [2].

> (4) *Balzám má **zmírňovat** a **předcházet otokům** v oblasti očí* [...]. (*The balm is supposed to relieve and prevent swelling in the eye area* [...].)

3 Data collection

We worked with Sketch Engine tool to collect the data, choosing the corpus czTenTen17 [2] as the source of linguistic material for the dataset. We searched the corpus with CQL queries focusing on structures containing verb coordination with specific context restrictions.

```
1. [tag="k1.*"][tag="k5.*"][word="nebo|a"][tag="k5.*"]
[tag="k1.*"]
2. [tag="k1.*"][tag="k5.*"][word="nebo|a"][tag="k5.*"]
[tag="k7.*"][tag="k1.*"]
```

The first two CQL queries seek after structures where the immediate context, i.e. the 1st position by KWIC [6], contains a noun on the left and either a noun or a preposition and a noun on the right. Furthermore, we removed passive forms from the search by the negative filter because, in such structures, the object moves to the subject position and the representation of the noun-verb relation changes from a government to an agreement.

```
3. [tag="k1.*c1.*"][word=".*" & tag!="kI.*"]{0,3}[tag="k5.*"&
tag!="k5.*mN.*" & lemma!="být"][word="nebo|a"][tag="k5.*"&
tag!="k5.*mN.*" & lemma!="být"][word=".*" & tag!="k[157I].*"]
[word=".*" & tag!="k[157I].*"]{0,1}[word=".*"&
tag!="k[157I].*"]{0,1}[tag="k1.*"] within <s/>
```

```
4. [tag="k1.*c1.*"][word=".*" & tag!="kI.*"]{0,3}[tag="k5.*"&
tag!="k5.*mN"][word="nebo|a"][tag="k5.*" & tag!="k5.*mN"]
[word=".*" & tag!="k[157I].*"][word=".*" & tag!="k[157I].*"]
{0,1}[word=".*" & tag!="k[157I].*"]{0,1}[tag="k7.*"]
[tag="k1.*"] within < s/>
```

The third and fourth CQL queries seek after verb coordinations where the immediate context, i.e., positions 1–3 from KWIC [6], contains a noun in nominative on the left, and a noun or preposition besides a noun on the right side. Within the immediate context on the left, we removed punctuation by the negative filter, and on the right side, we removed prepositions, verbs and punctuation on positions 1–3.

4 Linguistic data preprocessing for a manual annotation

To build a gold-standard annotated dataset, we used the web-based text annotation tool Brat [7] that supports, for instance, two basic types of annotations. It allows adding a label to a specific word (text span annotations) and adding relations among words in a sentence (relation annotations).

4.1 Data preprocessing

Since developing our text markup methodology for annotations in Brat would be inefficient, we took advantage of the UDPipe 2 [8] that works with CoNLL-U formatted files. It parses the input text file into sentence segments, giving each word a set of features (lemma, part-of-speech tag, morphological tag, dependency relation).

For the conversion of the UDpipe 2 (CoNLL-U) format to the standoff format for Brat, we use the ConllXtostandoff.py program [9] that creates .txt files containing the original sentences and .ann files with annotations from the CoNLL-U format, which Brat graphically displays.

Brat enables text annotation editing if particular labels are defined in the configuration files. We designed a script makeconffiles.py that extracts a required set of files (annotation.conf, tools.conf, visual.conf) from the output of UDPipe 2.

UDPipe 2 uses the positional morphological tag system [10], universal dependency tags [11] and universal dependency relations [12], which are developed for consistent grammar annotations across many languages.

The annotation.conf file defines universal positional tags (NOUN, ADJ, ADV and other) at the text span annotation level and universal dependency relations (*nsubj*, *obj*, *conj* and other) at the relation level. For our purposes, the essential dependency relation is coordination (*conj*). In dependency relations, it is a relation between two elements connected by conjunctions *and*, *or*. The head of this relation is the first conjunct, while the other elements depend on it [13].

4.2 Replacing the relation *conj* between coordinated verbs

We rename a syntactic relation *conj* in the coordinations, where both conjuncts have a common argument to *coordComArg*. If the argument does not grammatically correspond to the syntactic pattern of one of the conjuncts, we mark this defective structure as zeugma with label *coordZeug*. If conjuncts do not share any part of the sentence (except subject), we label the relation as *coordSent*. The original *conj* tag represents other types of coordinations.

4.3 The standard dataset – statistics

The manually tagged dataset consists of 2610 segments sorted by the number of ten to 261 files. One segment is a part of a sentence as parsed by the UD Pipe 2 tool. We randomly pick sentences from language material that we gained from corpus czTenTen17 [2]. The resulting statistics shows table 1.

Table 1: Statistics of the manually annotate dataset

Data set statistics	Count
Segments	**2610**
CoordComArg	**682**
CoordSent	**1506**
CoordZeug	**22**

5 Annotation automatization

Manual annotation of raw text is a time-consuming process, and the usage for machine learning requires thousands of annotated cases of the desired structures. We decided to design a script for relabeling relations in the UD Pipe 2 output to speed up annotations. We defined rules for the detection of zeugma and verb coordinations with or without a common argument.

5.1 Rules drafts

Based on the manual annotations experiences, the first step was to formulate theoretical rules for the automatized retrieval of the *coordComArg* and *coordZeug* structures. In addition, we define the *coordSent* relation as any other verbal coordination that the rules for distinguishing *coordComArg* and *coordZeug* do not cover. We describe these rules in the following two subsections.

CoordComArg This rule defines the verb coordination with a possible common argument. The prerequisite for the *coordComArg* structure is an identical valency of the verbs (see coordination below (5) [2]).

> (5) *Lada v současnosti* ***vyvíjí a vyrábí své vlastní automobily****.* (*Lada is currently developing and producing its cars.*)

To that purpose, we need to create a list of possible valency complements from a valency lexical database and, for each complement, a list of verbs that can bind with it. We assume that if two coordinated verbs are in the same list and simultaneously have suitable complements in the neighbouring context but not in their own, we consider this complement as shared.

CoordZeug We assume that verbs yoked by another sentence element in such structures require a different valency complement (6) [2].

> (6) *Analyzujte, jak organizace* ***rozhoduje a komunikuje změny.*** (*Analyze how the organization makes decisions and communicates changes.*)

We will use the list of valency complements again to follow the assumption that zeugma will most likely arise in coordinations with different verbal valency patterns if, in the context of the first, or the second verb, in the sentence, the appropriate complement is not found.

5.2 Implementation of the rule drafts

We generated a dictionary from the lexical database of Czech verb valency frames VerbaLex, [14] where the keys of the dictionary consist of any first obligatory complements of verbs in the database. The values of these keys contain lists of verbs that can have such complements according to the database. We saved the data structure in a .json file.

Further, we wrote a preprocess_relations.py script that takes as input the UDPipe 2 output in ConLL-U format. The program first goes through the input file and searches for verb coordinations, storing them in a list of tuples containing the ids of the sentences and verbs and lemmas.

The program also stores important sentence features for each word (word id, lemma, word type, tag, binding position, dependency relation) in a dictionary where the key is the sentence id (sent_id) and the value is a list of tuples.

The script does not handle reflexive verbs, as it is impossible to determine without any other rule whether the clitic ”se” is part of the verbal or noun phrase from the context of the sentence.

Program stores a context for each coordination in the list of two tuples that represent the left and right context. The context span is regularly five positions from each verb (KWIC <-5, 5>). The tuples store the numbers of the cases of such sentence positions where the nouns PRON (pronoun), NOUN, DET (determiner) and PROPN (proper noun) occur.

Coordinations are further processed using a dictionary generated from VerbaLex. Each verb obtains the list of arguments based on the dictionary. If the verbs can have the same argument structure, accordingly to the dictionary, and do not have a suitable complement in their context, they are stored with the ids to the list of common argument verb coordinations.

Similarly, we handle zeugmatic coordinations. If the verbs do not occur in the same list in the VerbaLex dictionary, and at the same time one of the verbs does not have a suitable binding in the context, the sentence id and verb id are saved into the list.

The output of the whole program is a newly processed CoNLL-U format file, renaming original *conj* relation to *coordComArg* and *coorZeug* according to the created lists. The *coordSent* relation matches the coordinations that do not cover the lists for *coordComArg* and *coorZeug*.

6 Comparing automatic and manual annotations

We tested the annotation preprocessing program on the dataset that we manually annotated in Brat, which covers mainly grammatically correct structures and on the dataset created for evaluating zeugma detection [15], where the zeugma occurs in significantly higher numbers. Table 1 and table 3 illustrate the evaluation of the program.

Table 2: Evaluation of automatical annotation on dataset focused on correct verb coordination. CoordCA – coordComArg, coordSe – coordSent, coordZe – coordZeug.

		Actual					
		coordCA	**coordSe**	**coordZe**	**Precision**	**Recall**	**F-score**
Predicted	**coordCA**	396	279	6	58,15 %	55,70 %	56,90 %
	coordSe	298	1106	7	78,38 %	73,05 %	75,62 %
	coordZe	17	129	11	7,01 %	45,83 %	12,15 %

We gained 58,15 % precision in detecting the common argument of two verbs and 55,70 % coverage based on the data. According to these results, we can assume that the program could significantly speed up the annotation. We could refine the program with more sophisticated going through the VerbaLex database and more accurate processing of context coordination (e.g., it does not consider whether punctuation is present in the context so that the program may consider a noun in another sentence as the verb object). Furthermore, the absence of several verbs (for example, *ignore*, *overthrow*) in VerbaLex causes false negatives and the failure to process coordination.

The rules for the detection of the zeugma proved to be ineffective. Most of the false-positive cases is caused by naive searching of the coordinations context and also by ellipses. In sentence seven [2], we see a typical example of a mislabeled zeugma. According to VerbaLex, the verbs *depart*, *leave* have obligatory complements that do not match. However, the verb *depart* has no complements in its context, so the coordination is evaluated as a zeugma.

> (7) *Vojáci* ***odcházejí a nechávají Achilla...*** (*The soldiers are departing and leaving Achilles.*)

Based on the results of the automatic annotations, we found that some coordination went unnoticed in the dataset with manual annotations. With the annotation preprocessing, we managed to get better results for coordinations with a common argument compared to the original data, as shown in Table 3.

Table 3: Statistics actualization of the dataset focused on verb coordinations with a shared argument

Dataset with preprocessed annotations	Count	Manualy annotated dataset	Count
Segments	**2610**	Segments	**2610**
CoordComArg	**1551**	CoordComArg	**1506**
CoordSent	**712**	CoordSent	**682**
CoordZeug	**22**	CoordZeug	**22**

As we see in Table 4 the precision of zeugma recognition improved many times on the dataset focused mainly on the zeugma phenomenon. However, this is a result of the unbalance of the dataset. Therefore, it might be beneficial to merge the two datasets; the rule evaluation results could then be more reliable. We could increase the recall of the rules by including reflexive verbs in the preprocessing and by designing a special rule to recognize such coordinations that may have the same binding in specific contexts.

The evaluation of the rules on the zeugma-focused dataset showed decrease in precision and recall scores for the *coordComArg* and *coordSent* relations rules. In the dataset where ungrammatical constructions are much more frequent, the

Table 4: Evaluation of automatical annotation on dataset focused on zeugma phenomenon. CoordCA – coordComArg, coordSe – coordSent, coordZe – coordZeug.

		Actual					
		coordCA	**coordSe**	**coordZe**	**Precision**	**Recall**	**F-score**
Predicted	**coordCA**	508	161	422	46,56 %	52,59 %	49,39 %
	coordSe	436	563	305	43,17 %	67,91 %	52,79 %
	coordZe	22	105	282	68,95 %	27,95 %	39,77 %

simple processing of valency frames from VerbaLex and naive passing through the context of verb coordination might have been more evident.

7 Summary and future work

This paper presented approaches we applied for building a dataset focused on coordinate structures of two verbs. The aim is to create a gold-standard dataset that can be used for training and testing a classifier for zeugma and verb coordinations with a shared argument recognition using machine learning methods.

We described the possibilities of speeding up the manual annotation process with automatic preprocessing, which could help create an extensive dataset with thousands of positive cases.

The outlined preprocessing showed promising results on tested data. However, annotation accuracy can be increased by improved coordination context managing, additional inclusion of reflexive verbs in the processing, and refined work with the valency frame database.

Therefore, we will continue editing and expanding the dataset in terms of content using the presented methods.

Acknowledgements. This work was supported by the project of specific research *Využití strojového učení při detekci společného argumentu v koordinovaných strukturách* (The application of machine learning methods to shared argument detection in verbal coordination structures); project no. MUNI/A/1184/2020).

References

1. Panevová, J., Gruet Škrabalová, H.: Elipsa. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), CzechEncy - Nový encyklopedický slovník češtiny. `URL:https://www.czechency.org/slovnik/ELIPSA` (2017)
2. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen corpus family. In: 7th International Corpus Linguistics Conference CL. pp. 125-127. (2013)
3. Lopatková, M., Mírovský, J., Kuboň, V.: Gramatické závislosti vs. koordinace z pohledu redukční analýzy (in Czech, Grammatical dependencies vs. coordination

from the perspective of reduction analysis). In: V. Kůrková et al. (Eds.): ITAT 2014 with selected papers from Znalosti 2014, CEUR Workshop Proceedings Vol. 1214, Praha, Univerzita Karlova. (2014) 65

4. Karlík, P., Gruet Škrabalová, H.: Koordinace (in Czech, Coordination). In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), CzechEncy - Nový encyklopedický slovník češtiny. `https://www.czechency.org/slovnik/KOORDINACE` (2017)
5. Karlík, P.: Zeugma. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), CzechEncy - Nový encyklopedický slovník češtiny. `https://www.czechency.org/slovnik/ZEUGMA` (2017)
6. Cvrček, V.: Kvantitativní analýza kontextu (in Czech, Quantitative context analysis). Praha, Nakladatelství Lidové noviny. Studie z korpusové lingvistiky (2013) 25
7. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S. and Tsujii, J.: brat: a Web-based Tool for NLP-Assisted Text Annotation. In: Proceedings of the Demonstrations Session at EACL 2012. (2012)
8. Straka, M.: UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In: Proceedings of CoNLL 2018: The SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Stroudsburg, PA, USA,(2018) 197-207
9. Auffarth, B., Pyysalo, S, Ninjin: ConllXtostandoff: Script to convert a CoNLL X tabbed dependency tree format. `https://github.com/nlplab/brat/blob/master/tools/conllXtostandoff.py` (2006)
10. Hajič, J.: Popis morfologických značek – poziční systém (in Czech, Description of morphological tags - positional system). `http://ucnk.ff.cuni.cz/doc/popis_znacek.pdf` (2000)
11. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of LREC. `https://aclanthology.org/L12-1115/` (2012)
12. de Marneffe, C.-M., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C. D.: Universal Stanford Dependencies: A cross-linguistic typology. In: Proceedings of LREC. (2014)
13. Universal Dependencies contributors: Introduction. `https://universaldependencies.org/u/dep/conj.html` (2021)
14. Hlaváčková, D., Horák, A.: VerbaLex - New Comprehensive Lexicon of Verb Valencies for Czech. In: Computer Treatment of Slavic and East European Languages. p. 107-115, 6 pp. Bratislava, Slovakia: Slovenský národný korpus (2006)
15. Medková, H.: Automatic Detection of Zeugma. In: Horák, A., Rychlý, P., Rambousek, A.: Proceedings of the Fourteenth Workshop on Recent Advances in Slavonic Natural Languages Processing, Brno: Tribun EU (2020) 79-86.

DMoG: A Data-Based Morphological Guesser

Vojtěch Kovář[1,2], Pavel Rychlý[1,2]

[1] Natural Language Processing Centre
Faculty of Informatics, Masaryk University, Brno, Czechia
[2] Lexical Computing
Brno, Czechia

`{xkovar3,pary}@fi.muni.cz`
`name.surname@sketchengine.eu`

Abstract. We present a novel corpus-based approach to lemmatization of unknown words. The tool learns affix patterns from annotated data, and based on these patterns, it predicts other word forms that should be present in the corpus. A lemma candidate then comes from the pattern whose predictions are really found in the corpus.
We present a prototype implementation and an initial evaluation on Czech, which shows promising results.

Keywords: Lemmatization · Morphological guesser · Morphological analysis · Morphological guessing

1 Introduction

Lemmatization of natural languages is the process of assigning a lemma (base form) to each word in the input text. Typically, it is solved by a look-up in a large database of all possible word-lemma or word-tag-lemma combinations.

However, there are always words missing in the database, so-called out-of-vocabulary (OOV) words: rare words, neologisms etc. In other cases, namely in low-resourced languages, there is no large word-lemma database available. In these cases, a morphological guesser is needed which suggests lemmas and/or parts-of-speech for OOV words.

In this paper we present a novel approach to the problem of morphological guessing based on checking guesser's predictions against corpus data. We also present a prototype implementation which is so far only limited to guessing lemmas (not tags) based on suffixes – on the other hand, the tool is extremely simple (less than 120 lines of Python code) and extensions are straightforward. Also, for some languages (including Czech, our testing language), this may already be useful and sufficient.

2 Related work and its drawbacks

Existing solutions which include [1] or [2] rely on longest affix matching between a particular OOV word and patterns learned from an available database.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 135–138, 2021. © Tribun EU 2021

In certain contexts, this leads to wrong lemma candidates that sound funny to native speakers, such as the following output for a few Czech OOV words from [1]:

```
buřtguláš      buřtgulat      k5eAaImIp2nS,k5eAaPmIp2nS
knedlo         knednout       k5eAaPmAgNnS
flash          flasha         k1gFnPc2
groupe         groupat        k5eAaPmIp3nS
nVidia         nVidium        k1gNnSc2
komorbiditou   komorbiditý    k2eAgFnSc7d1
```

In all cases except the last one, the lemma should be the same as the word form and the lemma proposed by the tool does not exist in Czech at all. The last case is a noun in instrumental (*comorbidity*) and its lemma should be *komorbidita*.

3 Corpus-based approach

In this paper we present a different approach. Our tool learns morphological patterns from available data as well, but the patterns represent declination schemata as a whole; and instead of matching an isolated OOV word and searching for longest affix match, it generates word forms that the particular pattern predicts (including the candidate lemma) and checks how many of them occur in the corpus.

For example, if *buřtguláš* has a lemma *buřtgulat* then it corresponds to a pattern which also predicts existence of the following word forms:

```
buřtgulat  buřtgulal
buřtgulám  buřtguláme
buřtguláš  buřtguláte
buřtgulá   buřtgulají
...
```

If we check this list against the corpus, we find out that the only existing word form is *buřtguláš* – so this is not a really good candidate, although the suffix indicates it might be a verb.

On the other hand, if it is a noun with lemma *buřtguláš*, then it corresponds to another pattern which predicts the existence of the following forms:

```
buřtguláš
buřtguláše
buřtguláši
buřtgulášem
...
```

Let's say 3 of these forms really occurred in the corpus (or corpus word list, respectively). Then we say this pattern is more suitable for this OOV word than the verb pattern, even if the common suffix is short or non-existent.

3.1 Patterns

A *pattern* in our understanding is a set of suffix pairs (*s1*, *s2*) where *s1* needs to be stripped from a word form and then *s2* needs to be added, to get a lemma. For example, the pattern for the verb schema mentioned above would contain

```
(-ám, -at)
(-áš, -at)
(-á, -at)
(-ál, -at)
(-ám, -at)
(-l, -t)
(-áme, -at)
(-áte, -at)
(-jí, -t)
...
```

This would be learned from many Czech verbs like *dělat, hledat* etc.

4 Implementation

Our prototype implementation consists of two Python scripts, `train.py` and `guess.py`. The first one reads a list of correct word-lemma pairs (obtained from manual annotation, morphological database, or a high-quality corpus) and saves the learned patterns into a so-called *model* (which is, however, just a set of patterns like the one above).

The `guess.py` script reads the model, together with an input word list generated from a corpus (i.e. not just isolated OOV words, but the complete corpus word list). For each of the words in the list, it tries to match the word suffixes, for each pattern from the model. If there is a suffix match, the tool generates all the potential word forms predicted by the pattern, and checks how many of them are there in the word list. The pattern who predicts the most existing lemmas wins the game, and its predicted lemma is returned as the result for the particular word.

5 Evaluation

As a preliminary evaluation, we trained the model on the word-lemma list of the manually disambiguated DESAM corpus [3], including only word-lemma pairs with frequency at least 5.

As testing data, we took the 40 most frequent OOV words from the csTenTen17 web corpus [4]. The results of our tool were as follows:

- correct lemmas: 36
- incorrect lemmas: 4
- accuracy: 90%

We have compared this result with the tool introduced in [1], on the same 40 words. Its results were as follows:

- correct lemmas: 26
- incorrect lemmas: 14
- accuracy: 65%

Although we admit that the testing set is very small and that it contained some noise (like a few frequent English terms used within Czech texts), the difference seems to be quite significant. Based on this result, we believe our DMoG prototype is worth further development, as well as a deeper research of the method itself.

6 Conclusions

We have introduced a new method for guessing lemmas for out-of-vocabulary words. We have explained the method and presented a prototype implementation, the DMoG tool. Although the current implementation only deals with lemmas and suffixes (and not prefixes, infixes and tags), it can be extended in a straightforward way, which is also the main goal of the future work.

Although the work itself, as well as the evaluation, are so far only preliminary, the tool shows promising results.

Acknowledgements. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

1. Šmerk, P.: Towards Czech morphological guesser. In Petr Sojka, A.H., ed.: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, Brno, Masarykova univerzita (2008) 1–4
2. Jongejan, B., Dalianis, H.: Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. (2009) 145–153
3. Pala, K., Rychlý, P., Smrž, P.: DESAM — annotated corpus for Czech. In: Proceedings of SOFSEM'97, Berlin, Springer (1997) 523–530
4. Suchomel, V.: csTenTen17, a recent Czech web corpus. In Aleš Horák, P.R., Rambousek, A., eds.: Proceedings of the Twelfth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2018, Brno, Tribun EU (2018) 111–123

Part IV

Text Corpora

When Word Pairs Matter

Analysis of the English-Slovak Evaluation Dataset

Michaela Denisová and Pavel Rychlý

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`{449884,pary}@mail.muni.cz`

Abstract. Cross-lingual word embeddings facilitate the transfer of lexical knowledge across languages, and they are mainly used for finding translation equivalents. Translation equivalents obtained in this way are usually evaluated with the help of ground truth dictionaries. However, the evaluation process, including the ground truth dictionaries, differs from model to model, impeding the correct interpretation of the results. Therefore, in this paper, we provide a thorough analysis of the English-Slovak ground truth dictionary and employ our analysis in evaluating two cross-lingual word embedding models. We show that word pairs choice is an important factor when accurately reflecting the model's performance.

Keywords: Cross-lingual word embeddings · Ground truth dictionary · Evaluation · English · Slovak

1 Introduction

In recent years, the popularity of cross-lingual word embeddings has risen among researchers due to their ability to connect meanings across languages. Cross-lingual word embeddings enable us to align the word vector representations from two or several languages into a single vector space where similar words obtain similar vectors [10]. In most cases, cross-lingual embedding models are evaluated via finding translation equivalents known as bilingual lexicon induction task [9,4,1]. In the bilingual lexicon induction task, translation equivalents are obtained from the aligned vector space through nearest neighbor search and then compared to the ground truth dictionaries. However, there is no united evaluation procedure agreed upon, and many authors consider different evaluation strategies, starting with different ground truth dictionaries, which causes inconsistencies between the stated results [5].

In this paper, we want to thoroughly analyze the English-Slovak dataset with 2,739 word pairs (1,500 English headwords) used as a ground truth dictionary to evaluate the MUSE model [4] and assign weight to each word pair accordingly. We aim to evaluate MUSE and VecMap [1] models with and without weighted word pairs to see how the model's performance changes. We think that current ground truth dictionaries used for evaluation may contain mistakes and

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 141–149, 2021. © Tribun EU 2021

irrelevant word pairs. Usage of such an evaluation dictionary can distort the actual model's performance.

The reason for our experiment is not to penalize the model when it does not find word pairs with lower weight, and we want the model to achieve higher accuracy when it includes word pairs with higher weight. Also, we believe that having a good quality evaluation dataset can reflect the model's performance more realistically and be the first step to a united evaluation procedure.

This paper is structured as follows. In Section 2, we describe MUSE and VecMap models. In Section 3, we analyze the English-Slovak dataset, and in Section 4, we use this dataset for MUSE and VecMap model evaluation. In Section 5, we offer concluding remarks.

2 Related Work

2.1 MUSE

The English-Slovak dataset we used for the analysis originates from the MUSE project. MUSE is an open-source cross-lingual word embedding model published by Facebook research in 2018. Except for the model, there are available pre-trained multilingual word embeddings aligned into shared vector space for 35 languages and ground truth evaluation dictionaries for 6 European languages in every direction and for 47 languages more from and to English. The model could be trained in a supervised [4] or unsupervised way [7]. For the supervised training, the Procrustes iterative alignment is used. The unsupervised method uses adversarial training and iterative Procrustes refinement.

In our experiments, we used supervised pre-trained multilingual embeddings for English and Slovak that are available in the MUSE library.[1]

2.2 VecMap

VecMap is an open-source cross-lingual word embedding model[2] released by Artetxe et al. in 2016. It provides four types of training: supervised [1], semi-supervised or identical training (relying on identical strings) [2], and unsupervised training [3]. For all of them, are required pre-trained monolingual word embeddings. Additionally, for semi-supervised and supervised training is necessary to have a training dataset from 25 up to 5,000 word pairs, respectively.

In this paper, we trained the model under strong supervision using the English-Slovak training dataset obtained from MUSE with 5,000 word pairs. Moreover, we used fastText monolingual embeddings [8] for English and Slovak in the training, downloaded from fastText library.[3]

---

[1] `https://github.com/facebookresearch/MUSE`

[2] `https://github.com/artetxem/vecmap`

[3] `https://fasttext.cc/`

3 Analysis of the Dataset

In the analysis, we considered three aspects that can influence the quality of an evaluation dataset. The first one was the frequency of given word pair in the parallel corpus. We obtained the frequencies for each word pair from the parallel English-Slovak corpus OPUS2 [11] via SketchEngine API [6]. The corpus contained approximately 8,000,000 sentences derived from 8,000 documents.

Logarithmic Zipf's curve of the obtained frequencies in Fig. 1 shows that most of the word pairs in the dataset had a lower frequency than 2,500.

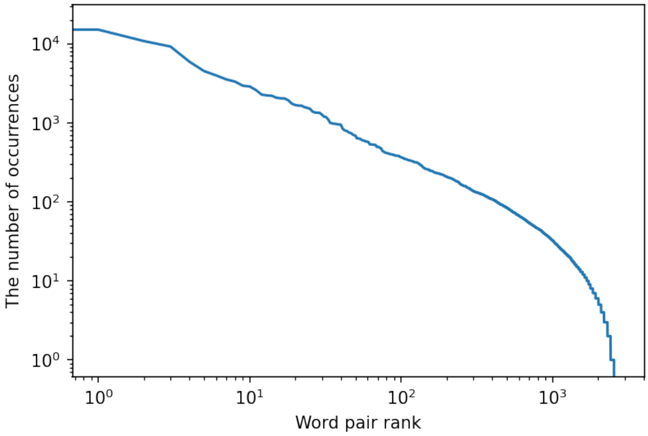


Fig. 1: Frequency distribution of each word pair in the parallel English-Slovak corpus OPUS2 represented by logarithm of Zipf's curve

In the following step of the analysis, we manually checked the word pairs, and according to the observed mistakes, we divided them into categories from A to J. The A category was for the correct translations, and the rest was for minor or major mistakes in the translations. For example, we found inflected word forms (*'compiled'*: *'zostavujú'*, *'advocacy'*: *'obhajobu'*), words translated with the same word that is not in Slovak (*'brook'*: *'brook'*), abbreviations (*'bbc'*: *'bbc'*), proper names (*'bruno'*: *'bruno'*) or even non-existing English words (*'wwe'*: *'mozeme'*), etc. Each category and its explanation are shown in Table 1.

The bar chart in Fig. 2 outlines how many word pairs were in each category. Given the graph, most of the word pairs received category A. However, the translation was not always the most frequent one (e.g., *'customer'*: *'odberateľ'*).

In the last step, we proposed our Slovak translation for each incorrect word pair in the categories from B to J. All word pairs in the A category kept their original Slovak translations. After annotating the English headwords with our Slovak translations, we measured the cosine similarity between word vector

Table 1: Categories, their description, weights and an example of a word pair from the respective category.

Category	Description	Weight	Example
A	correct translation	1	*'admit'* : *'priznať'*
B	inflected word form	0.80	*'advocacy'* : *'obhajobu'*
C	different part of speech	0.30	*'darkness'* : *'temné'*
D	translated as same non-Slovak word, abbreviations	0.20	*'bbc'* : *'bbc'*
E	proper names	0.20	*'bruno'* : *'bruno'*
F	synonym or incorrect translation	0.10	*'intensity'* : *'svietivosť'*
G	incomplete word pair	0.20	*'brigadier'* : *'brigádny'* (generál)
H	non-existing English word	0.10	*'wwe'* : *'mozeme'*
I	interjection	0.80	*'boom'* : *'bum'*
J	missing diacritics	0.60	*'joy'* : *'radost'*

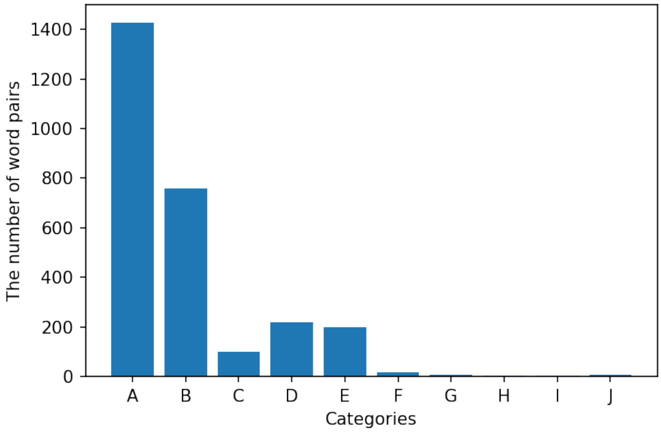


Fig. 2: The number of word pairs in each category.

representations of the original translation and our suggestion. To obtain these word vector representations, we used a pre-trained fastText word embedding model for the Slovak language. The results of this experiment are shown in Fig. 3.

3.1 Assigning weights

Given the described aspects, we assigned a weight to each word pair to reflect its relevance. Another reason was to increase the accuracy when the model finds word pairs with higher weight and not penalize the model for not including word pairs with a lower weight.

The weight was determined to be in the range between 0 to 1, so the first necessary step was to scale frequencies of the word pairs to the same range. However, as shown in Fig. 1, the word's frequency is inversely proportional to

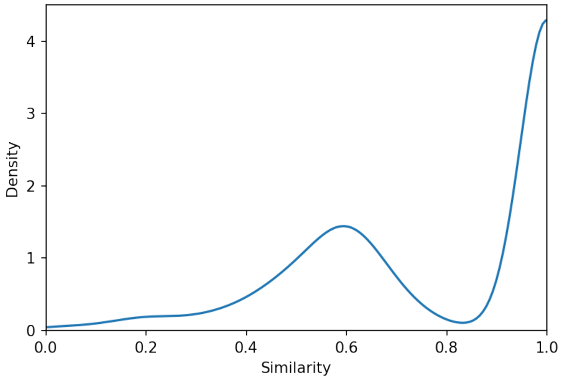


Fig. 3: Cosine similarity distribution from 0 to 1.

the word's rank, meaning that only a few word pairs have a very high frequency (the highest is 19,077), and the majority of the word pairs in the dataset have a frequency lower than 2,500. As a result, most word pairs would receive very small weight. The solution was to compute the logarithm of each weight first and then re-scale the numbers to the range between 0 to 1.

Furthermore, we added weights between 0 to 1 to each category, depending on whether the category represents a major or minor mistake. For example, category B or I was not considered a huge mistake, so it received higher weight while the weights for categories D and E were significantly lower. Categories, their explanations with an example, and assigned weights are shown in Table 1.

The cosine similarity was already in the range from 0 to 1, so it was not needed to process it.

Having frequencies scaled, weights for categories assigned, and cosine similarities computed, we multiplied these three values to obtain the weights for each word pair. Fig. 4. shows the overlapping histograms of weights distribution in each category.

However, the assigned categories and cosine similarity computed between the word vector representations of the original Slovak translation and proposed translation are subjective aspects. Thus we decided also to use only scaled frequencies (to the range from 0 to 1) obtained from the parallel corpora as weights for the word pairs when evaluating the models. The following sections discuss the results.

4 Evaluation

We chose models MUSE and VecMap, for the evaluation to see how the performance changes before and after applying weights on each word pair in the test dataset. We divided weights into two subcategories: first is weights computed

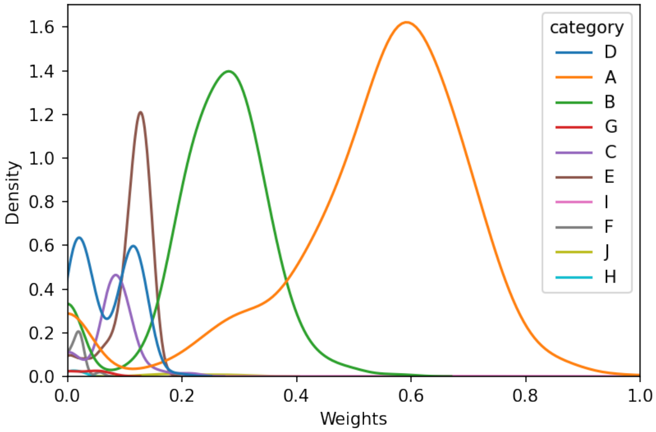


Fig. 4: Histograms of weights distribution in each category.

from weighted categories, frequencies, and cosine similarity, and the second one is scaled frequencies of the word pairs used as weights. Table 2 summarizes the results.

Table 2: The performance of Muse and VecMap models before and after applying weights and scaled frequencies used as weights on each word pair in the evaluation dataset.

	Without Weights	With Weights	Scaled frequencies
Muse (%)	30.41	**34.60**	32.82
VecMap (%)	38.15	48.43	**54.74**

Firstly, we downloaded from the muse library pre-trained word embeddings aligned into a single vector space for English and Slovak language. The English-Slovak evaluation dataset contained 2,793 word pairs and 1,500 English headwords, so we extracted the nearest neighbors of each English headword from the aligned vector space, depending on how many times the headword occurred in the evaluation dataset. For example, we extracted the first three nearest neighbors if there was an English headword with three different Slovak translations. Then we compared how many extracted word pairs using the muse model matched word pairs from the evaluation dataset. In the second evaluation, we included the weights from our analysis and scaled frequencies of the word pairs.

According to Table 2, the model's performance did not markedly change when using scaled frequencies as weights, but the numbers are slightly higher when considering weights from the analysis.

For the VecMap, we trained the model on English and Slovak FastText monolingual embeddings. The training was under strong supervision using 5,000 English-Slovak word pairs obtained from the muse training dataset. The result was embeddings for English and Slovak aligned to a single vector space. The evaluation part was the same as for the muse model. In comparison to the previous model, performance was significantly better when applying weights on each word pair. The best performance model achieved when considering only scaled frequencies as weights.

We examined and compared the word pairs that muse and VecMap models found through nearest neighbor search. muse looked up 294 word pairs from the evaluation dataset that VecMap was not able to find. Reversely, VecMap found 506 word pairs that muse did not include. Both models matched in 539 word pairs. Table 3 displays word pairs with the highest frequency and/or highest weight in which muse and VecMap models differ from each other.

Table 3: Comparison of the word pairs with the highest frequency (in hits per million) and/or highest weight that were found either by muse or VecMap model.

EN	SK	Frequency	Weight	Muse	VecMap
decrease	*zníženie*	**274**	**0.8709**	Yes	No
estonia	*estónsko*	42	0.7592	Yes	No
luxembourg	*luxembursko*	39	0.7555	Yes	No
euro	*eurá*	188	0.3957	Yes	No
vii	*vii*	254	0.1733	Yes	No
carefully	*starostlivo*	101	0.8115	No	Yes
decrease	*pokles*	253	0.8663	No	Yes
infection	*infekcia*	283	**0.8730**	No	Yes
hey	*hej*	1349	0.7728	No	Yes
tel	*tel*	**2384**	0.2000	No	Yes

5 Conclusion

Although applying weights in the evaluation of the muse model did not change the results remarkably, they helped to provide a more accurate picture of the VecMap model. VecMap outperforms the muse model in every evaluation discipline, and the evaluation proposes that VecMap is better when considering the most frequent word pairs in the parallel corpora.

Moreover, this analysis suggests that the choice of the word pairs and their frequency in corpus plays an important role in the evaluation and can reflect the model's performance more accurately.

Future work should focus on the analysis of the evaluation datasets for various language pairs. Especially we want to emphasize the morphologically rich languages to see to what extinct the inflected word forms influence the evaluation of the model's performance.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

1. Artetxe, M., Labaka, G., Agirre, E.: Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2289–2294 (2016), `https://aclanthology.org/D16-1250`
2. Artetxe, M., Labaka, G., Agirre, E.: Learning bilingual word embeddings with (almost) no bilingual data. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 451–462 (2017), `https://aclanthology.org/P17-1042`
3. Artetxe, M., Labaka, G., Agirre, E.: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 789–798 (2018), `https://arxiv.org/abs/1805.06297`
4. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. arXiv preprint arXiv:1710.04087 (2017), `https://arxiv.org/abs/1710.04087`
5. Glavaš, G., Litschko, R., Ruder, S., Vulić, I.: How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 710–721. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1070, `https://aclanthology.org/P19-1070`
6. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The sketch engine: ten years on. Lexicography pp. 7–36 (2014), `http://dx.doi.org/10.1007/s40607-014-0009-9`
7. Lample, G., Conneau, A., Denoyer, L., Ranzato, M.: Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043 (2017), `https://arxiv.org/abs/1711.00043`
8. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018), `https://arxiv.org/abs/1712.09405`
9. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. ArXiv **abs/1309.4168** (2013), `https://arxiv.org/abs/1309.4168`

10. Ruder, S., Vulić, I., Søgaard, A.: A survey of cross-lingual word embedding models. J. Artif. Int. Res. **65**(1), 569–630 (May 2019), `https://doi.org/10.1613/jair.1.11640`
11. Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012), `https://aclanthology.org/L12-1246/`

Transferability of General Polish NER to Electronic Health Records

Krištof Anetta and Mahmut Arslan

Natural Language Processing Centre,
Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
`xanetta@fi.muni.cz`, `xarslan@fi.muni.cz`

Abstract. This paper investigates the transferability of general Polish named entity recognition tools to the analysis of Polish health records. The tools, namely PolDeepNer2, spaCy's *pl_core_news_lg* pipeline and Spark NLP's *entity_recognizer_md* pipeline for Polish, were run on the *pl_ehr_cardio* corpus and their results were analyzed, paying special attention to their performance when processing these highly specific texts and to the applicability of the results in the healthcare domain. Even though the precision of PolDeepNer2 proved to be superior to both spaCy and Spark NLP, the paper concludes that without additional training, general named entity recognition tools for Polish have very limited use in the medical analysis of electronic health records. However, they could be helpful in partial tasks ranging from de-identification to entity disambiguation and discovery of mistyped entities or candidate entities that are not present in medical dictionaries.

Keywords: EHR · Electronic health records · Healthcare texts · NER · Named entity recognition · NLP · Natural language processing · Slavic languages · Polish · PolDeepNer2 · spaCy · Spark NLP

1 Introduction

In the past decade, NLP for healthcare, especially entity recognition, has been growing rapidly in the English-speaking world. However, low-resourced languages like Polish have been progressing much more slowly due to the combined effects of a lack of resources at every level of processing. The key disadvantage is the absence of a Polish UMLS translation - while English UMLS boasts more than 9 million terms [1], facilitating knowledge extraction, Polish only has around 50,000 terms in the MeSH subset, which is both too sparse and too general to be of use in health records. Until better Polish healthcare dictionaries are developed, researchers have the option to train deep learning entity recognition systems to find strings which are likely to be medical entities based on their features. As there are currently no benchmark tools for discovering Polish medical entities (notable work has been done by [2], but without generalizable search for new entities), this paper surveys the borderland between general entity recognition and healthcare entity recognition, trying to find out to what extent the

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 151–159, 2021. © Tribun EU 2021

Table 1: Mapping of entity categories

	PolDeepNer2	spaCy	Spark NLP
PER	nam_liv	persName	PER
ORG	nam_org	orgName	ORG
LOC	nam_loc nam_fac	placeName geogName	LOC
MISC	nam_eve nam_pro nam_adj nam_num nam_oth	date time	MISC

existing general Polish entity recognition systems can be ported to the healthcare domain.

When looking for named entities in Polish text, there are several options to consider [3], ranging from deep learning to dictionary-based approaches. In this paper, three recently updated options were chosen for comparison - PolDeepNer2 [4] with the KPWr n82 NER model [5] was chosen as the state-of-the-art, custom-made deep learning approach (categories were simplified for the statistics), spaCy's [6] *pl_core_news_lg* pipeline was chosen based on its effortless availability to any spaCy user, and Spark NLP's [7] *entity_recognizer_md* pipeline for Polish was chosen because of Spark NLP's noticeable presence in healthcare text processing - there are already clinical NLP models for English, German, and Spanish, which hints at potential future extensibility of Spark NLP's general Polish entity recognition into clinical entity recognition.

The analyzed corpus, *pl_ehr_cardio* [8], consists of more than 50,000 health records related to cardiology collected over 18 years at the Medical University of Silesia in Katowice, Poland. The corpus contains more than 23 million words.

2 NER Results in *pl_ehr_cardio*

In order to compare the results, a mapping between categories used by individual tools had to be decided. PER, ORG, LOC and MISC were chosen as the unifying categories with the mapping shown in Table 1. Table 2 compares the total counts and ratios of entities found in the corpus. Tables 3, 4, and 5 show entity statistics for the entire corpus processed by PolDeepNer2, spaCy, and Spark NLP, respectively.

Table 2: Total counts of entities in the *pl_ehr_cardio* corpus. Total word count of the corpus is 23,831,785.

	PolDeepNer2		spaCy		Spark NLP	
all	**725,198**		**965,225**		**3,428,457**	
PER	170,969	23.6%	350,749	36.3%	381,543	11.1%
ORG	119,321	16.5%	248,115	25.7%	502,457	14.7%
LOC	21,026	2.9%	78,888	8.2%	1,350,885	39.4%
MISC	413,882	57.1%	287,473	29.8%	1,193,572	34.8%

3 Performance Analysis

3.1 Analyzed Sample Characteristics

The sample chosen for manual analysis consisted of a pseudo-random selection of 17 patient records totaling 9382 words, evenly distributed across the 18-year timespan of the *pl_ehr_cardio* corpus. Table 6 summarizes the precision achieved by individual tools, in total and per category. The MISC category is not evaluated because it has a different meaning for each tool and its boundaries are fuzzy - furthermore, the status of a named entity is especially difficult to establish in medical terminology.

3.2 PolDeepNer2

PolDeepNer2 identified 193 named entities in the analyzed sample. It was the smallest number of entities of all the tools, but they were identified with significantly greater precision.

Names of people Within the sample chosen for analysis, 100% (54/54) of what PolDeepNer2 identified as names of people was correct, even though in most

Table 3: PolDeepNer2 statistics for entities. The ◁ symbol separates values for the minimum, average and maximum number of entities per the specified text block.

per	any entity	PER$_{\text{son}}$	ORG$_{\text{anization}}$	LOC$_{\text{ation}}$	MISC$_{\text{ellaneous}}$
sentence	0 ◁ 2.1 ◁ 32	0 ◁ 1.3 ◁ 11	0 ◁ 1.3 ◁ 12	0 ◁ 1.2 ◁ 13	0 ◁ 2.4 ◁ 31
paragraph	0 ◁ 4.0 ◁ 92	0 ◁ 2.0 ◁ 33	0 ◁ 1.6 ◁ 36	0 ◁ 1.4 ◁ 13	0 ◁ 4.1 ◁ 67
epicrisis physicalexam	0 ◁ 2.7 ◁ 38	0 ◁ 1.2 ◁ 8	0 ◁ 1.4 ◁ 10	0 ◁ 1.2 ◁ 6	0 ◁ 2.3 ◁ 24
epicrisis recommendation	0 ◁ 3.4 ◁ 25	0 ◁ 1.1 ◁ 8	0 ◁ 1.7 ◁ 11	0 ◁ 1.6 ◁ 12	0 ◁ 3.0 ◁ 21
interview onset	0 ◁ 5.8 ◁ 92	0 ◁ 1.4 ◁ 11	0 ◁ 1.8 ◁ 36	0 ◁ 1.4 ◁ 13	0 ◁ 5.2 ◁ 67
interview physicalexam	0 ◁ 2.3 ◁ 76	0 ◁ 2.1 ◁ 33	0 ◁ 1.0 ◁ 9	0 ◁ 1.3 ◁ 13	0 ◁ 1.4 ◁ 44
document	0 ◁ 9.5 ◁ 101	0 ◁ 2.4 ◁ 33	0 ◁ 2.5 ◁ 38	0 ◁ 1.6 ◁ 14	0 ◁ 6.9 ◁ 74

Table 4: Spacy statistics for entities. The ◁ symbol separates values for the minimum, average and maximum number of entities per the specified text block.

per	any entity	PER$_{\text{son}}$	ORG$_{\text{anization}}$	LOC$_{\text{ation}}$	MISC$_{\text{ellaneous}}$
sentence	0 ◁ 2.0 ◁ 70	0 ◁ 1.4 ◁ 16	0 ◁ 1.1 ◁ 11	0 ◁ 1.0 ◁ 5	0 ◁ 2.4 ◁ 58
paragraph	0 ◁ 5.0 ◁ 110	0 ◁ 2.7 ◁ 57	0 ◁ 1.8 ◁ 42	0 ◁ 1.7 ◁ 24	0 ◁ 4.0 ◁ 72
epicrisis phys.exam	0 ◁ 3.3 ◁ 50	0 ◁ 1.4 ◁ 14	0 ◁ 1.6 ◁ 12	0 ◁ 1.1 ◁ 6	0 ◁ 2.6 ◁ 37
epicrisis recomm.	0 ◁ 2.8 ◁ 25	0 ◁ 2.2 ◁ 16	0 ◁ 1.3 ◁ 9	0 ◁ 1.1 ◁ 4	0 ◁ 3.0 ◁ 21
interview onset	0 ◁ 6.5 ◁ 110	0 ◁ 2.1 ◁ 24	0 ◁ 2.2 ◁ 42	0 ◁ 1.4 ◁ 10	0 ◁ 4.6 ◁ 62
interview phys.exam	0 ◁ 4.9 ◁ 110	0 ◁ 3.1 ◁ 57	0 ◁ 1.6 ◁ 17	0 ◁ 2.0 ◁ 24	0 ◁ 6.3 ◁ 72
document	0 ◁ 12.1 ◁ 136	0 ◁ 4.7 ◁ 65	0 ◁ 3.6 ◁ 45	0 ◁ 2.2 ◁ 24	0 ◁ 5.7 ◁ 89

cases these names were parts of the names of medical examinations, conditions, and methods named after their discoverers or inventors ("objaw **Chełmońskiego / Blumberga / Goldflamma / Babińskiego**", "choroba **Buergera**", "metodą **Holtera**"). This unintended capability proves especially useful in cardiology where discoverer-based medical concept names are common. With some additional rule-based evaluation on top of PolDeepNer2's person name recognition, it could be a useful addition to a Polish healthcare text processing system.

Names of organizations Medical organization names were more difficult for PolDeepNer2, but it still fared quite well - in the analyzed sample, 81.8% (36/44) of strings identified as organization names were in fact names of organizations or individual departments and offices of those organizations ("**Poradni Kardiologicznej i Diabetologicznej**", "**Szpitala w Tychach**", "**Szpitala w Świętochłowicach**", "**Oddziału Intensywnej Terapii z Nadzorem Kardiologicznym**"). Almost all of the errors occurred in the most difficult kind of organization names - capitalized abbreviations. Apart from surprising success with some instances ("**OAITK zNK**", "**OITK**", "**POChP**", "**MIC**", "**POZ**"), there were some non-organization abbreviations that slipped in ("**LAD**", "**UKG**",

Table 5: Spark NLP statistics for entities. The ◁ symbol separates values for the minimum, average and maximum number of entities per the specified text block.

per	any entity	PER$_{\text{son}}$	ORG$_{\text{anization}}$	LOC$_{\text{ation}}$	MISC$_{\text{ellaneous}}$
sentence	0 ◁ 1.9 ◁ 82	0 ◁ 1.2 ◁ 10	0 ◁ 1.2 ◁ 15	0 ◁ 1.3 ◁ 49	0 ◁ 1.8 ◁ 36
paragraph	0 ◁ 19.0 ◁ 536	0 ◁ 2.7 ◁ 38	0 ◁ 6.1 ◁ 144	0 ◁ 8.1 ◁ 214	0 ◁ 7.3 ◁ 178
epicrisis phys.exam	0 ◁ 2.7 ◁ 536	0 ◁ 1.4 ◁ 11	0 ◁ 1.4 ◁ 19	0 ◁ 10.8 ◁ 214	0 ◁ 2.3 ◁ 24
epicrisis recomm.	0 ◁ 5.3 ◁ 52	0 ◁ 1.7 ◁ 10	0 ◁ 2.1 ◁ 17	0 ◁ 2.3 ◁ 16	0 ◁ 3.4 ◁ 32
interview onset	0 ◁ 11.7 ◁ 272	0 ◁ 2.2 ◁ 31	0 ◁ 2.2 ◁ 29	0 ◁ 5.2 ◁ 117	0 ◁ 6.0 ◁ 104
interview phys.exam	0 ◁ 2.3 ◁ 76	0 ◁ 3.3 ◁ 38	0 ◁ 8.2 ◁ 144	0 ◁ 1.3 ◁ 13	0 ◁ 10.3 ◁ 178
document	0 ◁ 40.5 ◁ 567	0 ◁ 5.0 ◁ 38	0 ◁ 8.8 ◁ 145	0 ◁ 15.9 ◁ 218	0 ◁ 15.6 ◁ 188

"**POWLOK**", "**EKG**", "**LOC**"), likely due to the notorious syntactical insufficiency of health records that confused the contextual classifier.

Even though this might raise a suspicion that PolDeepNer2 chose these abbreviations superficially, based on the capitalization of all of their letters, it proves to be unfounded upon closer analysis - there were more than 300 capitalized abbreviations in the sample and only 12 of those were recognized as organization names, demonstrating the high specificity of PolDeepNer2's criteria.

In addition to the above, PolDeepNer2 was able to identify incomplete references to organizations ("**Kliniki**") and recognize an entity in spite of an error in a crucial noun ("**Kliniii Chirurgii Ogólnej i Naczyń**").

Names of locations In the analyzed sample, PolDeepNer2 only identified 1 occurrence of a location, which is not enough to evaluate its performance. This occurrence was labeled incorrectly, as it was a general reference to organizations ("w **Poradnich**") the syntactical use of which resembled a geographical name.

Miscellaneous names Miscellaneous is perhaps the most interesting category, since it has the potential to discover names that are actually relevant for medicine. PolDeepNer2 found 94 miscellaneous names, further divided into 16 product names, 9 event names, and 69 "other" names. Of the product names, 68.8% (11/16) can be considered correct, including 9 medicine names (e. g. "**Biosotal**","**Mixtrad**", "**Encorton**", "**Theovent**", "**Pentohexal 600**") and 2 device names ("w **Holterze**", "**EKG**"). Of the event names, 44.4% (4/9) were correct, identifying 2 heart attacks ("**NSTEMI**", "**Przebyty udar**") and 2 medical procedures (e. g. "**POBA**"). Errors in the product and event categories resulted from incorrectly labeling capitalized abbreviations with insufficient syntactical context, namely 100% (10/10) of errors were strings either entirely composed of capital letters and numbers or including a capitalized non-word substring (e. g. "**PTCA LAD**", "**Stan po POBA**", "**R57**").

The "other" category is more difficult to evaluate because almost anything in health records can be considered an entity, even though rarely a proper name. Of the 69 strings labeled in this way, there were 16 additional medicine names

Table 6: Performance comparison for commensurable categories. Precision was manually evaluated on a subset of records.

	PolDeepNer2		spaCy		Spark NLP	
all	**90.9%**	**90/99**	**40.3%**	**104/258**	**7.6%**	**59/780**
PER	100%	54/54	41.1%	53/129	34.4%	45/131
ORG	81.8%	36/44	50.5%	51/101	6.1%	11/179
LOC	0%	0/1	0%	0/28	0.6%	3/470

(in addition to the ones identified as product names) and a varied collection of medical states, procedure names, and institution name abbreviations. 79.7% (55/69) of the "other" names were strings that were either capitalized or exhibited a different sign of being an abbreviation, such as including a number (e. g. "**CCS II**", "**WZWB**", "**DDD**", "**Ao-OM2**", "**TILT**"). On the one hand, these matches seem to be highly relevant for medicine, but on the other hand, since the system has no idea what it has found, significant further processing or approach hybridization would be required to turn these discoveries into knowledge in big data.

3.3 spaCy

spaCy's *pl_core_news_lg* pipeline identified 403 entities in the analyzed sample.

Names of people spaCy identified 129 strings as names of people, but only 41.1% (53/129) were actual names, and they were exclusively the names of medical concepts named after their inventors, the very same ones that were described in the PolDeepNer2 section. 17.1% (22/129) were incorrectly labeled medicine names that probably confused the system by their capitalized first letter. Most of the remaining errors were standard words, often describing a body part or a characteristic looked for in the examination (e. g. "**TKANKA**", "**ODGLOS**", "**Wątroba**", "**Tony**"). Interestingly, there were cases where the first letter was not even capitalized ("**ablacją**", "**tężcowa**").

Names of organizations 101 strings were labeled as organization names, however, only 50.5% (51/101) were truly referring to organizations and their individual departments and offices. Similar to the names of people, the errors included 10 medicine names and a mix of regular words relating to medical examinations ("**Uczulenia**", "**stentem**", "**TARCZYCA**", "**Cholesterolu**")

Names of locations PolDeepNer2 already indicated that health records are not rich in location names and this was the case for spaCy as well. It identified 28 strings as names of locations, of which 0% (0/28) were correct in the proper, narrow sense of what a location is. There were, however, 15 instances of locations on the body ("**GRANICE DOLNE PLUC**", "**Spojówki**", "**Tarczyca**"), resulting from a syntactical similarity which could prove useful in the analysis of body references in health records.

Miscellaneous names spaCy's miscellaneous category only includes dates and times mentioned in the text, and is therefore quite different from the same category in the other tools. The performance of spaCy in this particular task was decent and potentially useful for temporal marking of health records. In the analyzed sample, 145 strings were identified as date or time, of which 97.2% (141/145) were correct. Errors included mistakenly labeled use of numbers, e. g. drug dosage or measurements ("**1-0-0**", "BMI **21.08**").

3.4 Spark NLP

Spark NLP’s *entity_recognizer_md* pipeline for Polish proved to be overwhelmingly optimistic in its guesses. It found 1193 entities in the analyzed sample, 6 times as many as PolDeepNer2 and 3 times as many as spaCy, which was already too optimistic to start with.

Names of people Interestingly, despite being extremely liberal with other labels, Spark NLP identified 131 strings as names of people, a result very close to spaCy’s 129. 34.4% (45/131) of these strings correctly captured a personal name, but often included other words that did not belong with the name (“**Objaw Goldflama**”, “**Objawy Chełmońskiego**”), likely because of the capitalization of the neighboring words. Only 19.8% (26/131) were clean personal names.

Names of organizations Spark NLP’s performance on organizations was outright abysmal. Only 6.1% (11/179) of the found strings were truly referring to organizations. The most obvious error pattern was related to capitalization - in 69.8% (125/179) of strings identified as organizations, more than half of all characters were either capital letters or numbers, thus resembling abbreviations and company/institution names, even if they were regular Polish words (e. g. “**WYPOWIADA**”, “**SKORA**”, “**CZASZKA**”).

Names of locations Compared with PolDeepNer2’s 1 and spaCy’s 28, Spark NLP’s 470 results for location names sounds too good to be true, and it is. Only 0.6% (3/470) of the strings identified as location names were geographical locations. Interestingly, 29.6% (139/470) of the strings represented locations on or within the body (“**Gałki**”, “**Śledziona**”, “**Brzuch**”), which, if the precision improved, could be useful for health record analysis. While body location errors can be explained by syntactical similarity, another notable error pattern is more difficult to explain: 6.8% (32/470) of the identified strings were medicine names (“**Acard**”, “**Milurit**”, “**Tertensif SR**”) which often stand alone in the text, outside of sentence structure, and therefore there seems to be no reason to consider them location names apart from the capital letter at their beginning.

Miscellaneous names In short, the noise in this category renders the results unusable. The 413 identified strings were chosen for indecipherable reasons and they ranged from meaningless fragments (e. g. “**V**”, “**Po**”, “**(EF**”, “**-0-10j).**”) to regular words to abbreviations and codes. Capitalization and code-like nature seemed to matter, as 44.1% (182/413) of the strings were more than half capitals or numbers.

An interesting error in the miscellaneous category was the labeling of very long strings. 19.6% (81/413) of the strings identified as miscellaneous names were longer than 20 characters, 6.5% (27/413) were longer than 30 characters, and 1.9% (8/413) were longer than 40 characters. None of these longer strings was a proper name.

4 Conclusion

The tested named entity recognition tools were facing a highly improbable task and they met, and in the case of PolDeepNer2, exceeded the expectations set at the start. That said, if we were to ask the question whether existing general named entity recognition for Polish can render useful results for electronic health records, the answer is a clear no - even in the tasks they are relatively good at (PolDeepNer2's performance in names of people and organizations), recall is threatened by the syntactical poverty of health record text, and once the tools attempt to identify other types of entities, they no longer label them correctly, thus providing no information on how to handle them. In addition to all this, the basic entity categories that the models are looking for do not overlap well with what is relevant for medical science. Names of body parts, symptoms, and diagnoses do not fit anywhere, the often abbreviated names of procedures, even though sometimes identified as events, end up scattered amongst categories, and some, but not enough names of medicines are identified by PolDeepNer2 as products. Even with radically improved performance, the existing tools would not be looking for the relevant data in the first place.

Of course, this is an unfair question to ask, as these tools were never intended for such texts - their failure is expected and understandable. A more productive question is whether the existing tools could be useful with some additional training or as a part of a more complex processing pipeline, and here the results suggest a much more positive outlook - especially PolDeepNer2, apart from providing the obvious and highly demanded service of de-identification by finding personal names with great precision, might be able to enhance dictionary-based lookup techniques for medical entities by providing candidate entities that are either unknown to the lookup system or distorted by errors, or it could help disambiguate the meaning of previously identified entities by labeling them with their role. Additional training on medicine names could easily improve the recognition of product names, which could go beyond the available databases of medical products and identify alternative product names or even the medicinal use of products that are originally non-medical.

Research on Polish electronic health records is still in its infancy, but the rapid global development of transformer architectures together with Polish-specific research initiatives are quickly progressing towards their first successes in mining structured data from the cryptic, time-pressured writing produced in hospitals and doctors' offices.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. The Polish data was provided by the Medical University of Silesia in Katowice, Poland, with the help of Prof. Krystian Wita, MD PhD, Prof. Wojciech Wojakowski, MD PhD, Prof. Wacław Kuczmik, MD PhD, Tomasz Jadczyk, MD PhD, and Marcin Wita, MD PhD.

References

1. Névéol, A., Dalianis, H., Velupillai, S., Savova, G., Zweigenbaum, P.: Clinical Natural Language Processing in Languages Other Than English: Opportunities and Challenges. *Journal of Biomedical Semantics*, vol. 9, issue 1, pp. 1-13, 2018. https://doi.org/10.1186/s13326-018-0179-8
2. Dobrakowski, A.G., Mykowiecka, A., Marciniak, M., Jaworski, W., Biecek, P.: Interpretable Segmentation of Medical Free-text Records Based on Word Embeddings. *Journal of Intelligent Information Systems*, 2021. https://doi.org/10.1007/s10844-021-00659-4
3. Marcińczuk, M., Wawer, A.: Named Entity Recognition for Polish. *Poznan Studies in Contemporary Linguistics* vol. 55, issue 2, pp. 239-269, 2019. https://doi.org/10.1515/psicl-2019-0010
4. Marcińczuk, M., Radom, J.: A Single-run Recognition of Nested Named Entities with Transformers. *Procedia Computer Science*, vol. 192, pp. 291-297, 2021. https://doi.org/10.1016/j.procs.2021.08.030
5. Marcińczuk, M.: KPWr n82 NER model (on Polish RoBERTa base). CLARIN-PL digital repository, 2020. `http://hdl.handle.net/11321/743`
6. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python. 2020. https://doi.org/10.5281/zenodo.1212303
7. Kocaman, V., Talby, D.: Spark NLP: Natural Language Understanding at Scale. *Software Impacts*, vol. 8, 100058, 2021. https://doi.org/10.1016/j.simpa.2021.100058
8. Anetta, K.: Data Mining from Free-Text Health Records: State of the Art, New Polish Corpus. In: Horák, A. (ed.) *Proceedings of the Fourteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2020*, pp. 13-22. Tribun EU, Brno (2020). ISBN 978-80-263-1600-8

A Case Study of High-Frequency Dictionary Collocations in a Spoken Corpus

Maria Khokhlova

St Petersburg State University,
Universitetskaya emb. 7-9-11, 199034 St Petersburg, Russia
`m.khokhlova@spbu.ru`

Abstract. Linguists rarely focus their attention on spoken corpora to study collocations, but these resources can suggest valuable examples. This article discusses the adj-noun frequency collocations from the Russian collocation database that constitute a gold standard. The aim is to compare the usage of collocations on the material of the oral and written corpora. The results show that low frequencies characterize dictionary collocations, and in most cases, the occurrences are adjacent combinations that do not include other words.

Keywords: Collocations · Spoken corpora · Evaluation · Dictionaries · Russian language

1 Introduction

In numerous studies, MWEs, collocations, and other set phrases were considered on the material of exclusively written texts and mainly from the point of view of their frequency. Oral data remained outside the scope of these works, which can be objectively explained by small volumes of oral texts available to researchers until recently, as well as the laboriousness of their processing.

Our paper focuses on the following questions: 1) do high-frequency collocations collected from dictionaries occur in spoken texts? 2) do their frequencies differ from the ones in written corpora?

The paper is structured as follows. The Introduction presents the basic idea of the research. The next section provides a brief overview of the spoken corpora. Section 3 discusses the methods and relevant notions essential for the analysis. The next section examines the experiment results, while the conclusion ends the paper and offers future perspectives.

2 Spoken Russian Corpora

Spoken corpora are not as common as their written counterparts since their building is a difficult task. However, we cannot overestimate their importance while they provide valuable data. There are not so many projects for Russian that

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 161–166, 2021. © Tribun EU 2021

focus on collecting oral data. Most of the existing ones are of a small volume and were compiled for a particular task (for example, to study learners' speech).

The Spoken Corpus of Russian (SCR) is a part of the Russian National Corpus [14] and has various types of annotation (morphological and lexical features and textual information). It includes transcripts of recordings of public and private oral speech, as well as film transcripts, and comprises about 13.4 mln tokens.

The project "Night Dream Stories and Other Corpora of Oral Speech" gave birth to several spoken corpora [12]. The first one comprises stories about night dreams that were retold by children and teenagers; its volume is 14,000 tokens. The second corpus consists of 17 stories described by adult residents of Novosibirsk (from 19 to 70 years old) about exciting events in their lives (5,000 tokens). The last collection includes 40 stories presented by adults (from 18 to 60 years old) about funny incidents in their lives (10,000 tokens).

The Corpus of Russian Oral Speech was compiled to study the processes of speech perception by native speakers; its texts have spelling annotation, as well as acoustic and phonetic transcription [2]. Currently, its total volume goes beyond 22,000 tokens, representing different styles of speech: professional voice-over reading, reading by native speakers, spontaneous monologue speech, and children's speech.

The ORD Speech Corpus ("One Day of Speech") was built using the method of long hours monitoring [10]. It includes data from 128 speakers and more than 1,000 interlocutors representing different social groups in St Petersburg. The whole length of the recordings is 1,450 hours; their transcribed version reaches over 1 mln tokens.

3 Methods

The statistical patterns of collocability cannot be considered without linguistic parameters, which show the real usage of word combinations in texts. As reference data, we will focus on collocations obtained by us earlier (see, for example, [6]) and constituting the so-called "gold standard". From the Russian collocations database described in [5], we selected 50 items with different dictionary indices, i.e. which are present in explanatory and specialized dictionaries ([1], [3], [4], [7], [8], [9], [11], [13]). The first group has the dictionary index equal to 5, which means that five dictionaries describe these collocations. In contrast, the examples from the second group were found only in two dictionaries. We proceed from the fact that collocations from the first group show high frequency in lexicographic resources and hence are highly reproducible in speech. Both groups represent the adj-noun structural model. Further, we considered occurrences of these collocations in the SCR and the written disambiguated subcorpus of RNC (6 mln tokens).

In order to establish how native speakers recognize collocations, it is necessary to collect additional information about their usage in texts. These parameters include not only information already available about frequencies or parts of speech (that is, standard statistical values applied at the text or entire corpus level) but also previously unexplored parameters of the behavior of units, for example, at the clause level. We speculate that any semantic shift within a collocation (e.g., semantic non-compositionality) deals with features that may be inferred from corpus data. One of them is permeability, i.e., the ability of a collocation to be split by a foreign token in-between. Hence we will study the representation of this characteristic that can be found in corpus examples. We will consider not only adjacent bigrams but also their distance equivalents (for example, *polnaya svoboda* "complete freedom" and *polnaya i bezgranichnaya svoboda* "complete and unlimited freedom").

4 Results

4.1 Dictionary indices 5 and 2

The majority of collocations from the first group were found in specialized dictionaries. One item was described in explanatory dictionaries, namely, *zhguchiy bryunet* "burning brunette" and has idiomatic features. Among the considered examples, two nouns have more than one collocation, namely, *glubokaya tishina* "deep silence", *polnaya tishina* "complete silence", *bogatyy urozhay* "rich harvest", *vysokiy urozhay* "high harvest". The most frequent collocate is *glubokiy* "deep" (8 examples), while such adjectives as *zheleznyy* "iron", *ostryy* "sharp" and *polnyy* "complete, full" show 2 examples.

The results for the group with the dictionary index 5 are shown in Table 1 (absolute frequencies). We can note a low correlation between two distributions (0.36 according to the Spearman coefficient, $p > 0.05$). However, the frequencies are small and do not differ in the corpora ($V=80$ according to the Wilcoxon test, $p > 0.05$).

For distance n-grams, we searched up to five words between a node and a collocate (the last column in Table 1). The selected collocations show low permeability. The average frequency is 0.68 and 0.80 for spoken and written texts, respectively. The following cases exemplify the longest n-grams: *tverdaya, khotya i mgnovenno sozrevshaya uverennost'* "firm, albeit instantly ripe, confidence"; *polnoy i ravnoy dlya vsekh svobody* "full and equal freedom for all".

Table 2 presents absolute frequencies for the collocations registered in two dictionaries. More than half of collocations from this group had no examples in corpora. They tend to occur rarer than the collocations mentioned above. Long n-grams were not found with only four exceptions, that are trigrams, e.g. *dlinnaya avtomatnaya ochered'* "a long gun burst", *chrezmernoye issledovatel'skoye svimaniye* "excessive research attitude", *bol'shoy vas poklonnik* "a big fan of you" and *svezhaya nemetskaya gazeta* "a fresh German newspaper".

The results suggest that both corpora are not sufficient in their volume to study collocations. The collocations from the second group tend to occur only in their adjacent forms.

Table 1: Results for the dictionary index 5.

Collocation		Freq (SCR)	Freq (RNC)	Dist(SCR)
bogatyy urozhay	“rich harvest”	3	3	1
bol′shoy avtoritet	“great authority”	12	0	1
vysokiy urozhay	“high yield”	5	0	0
glubokaya blagodarnost′	“deep gratitude”	4	2	1
glubokoye vliyaniye	“deep influence”	0	2	0
glubokoye znaniye	“deep knowledge”	6	7	1
glubokiy interes	“deep interest”	1	3	0
glubokiy krizis	“deep crisis”	3	3	4
glubokaya tishina	“deep silence”	3	3	0
glubokoye ubezhdeniye	“deep refuge”	25	9	1
glubokoye chuvstvo	“deep feeling”	1	5	1
goryachaya lyubov′	“hot love”	6	1	1
grubaya oshibka	“big mistake, blunder”	12	5	0
zhguchiy bryunet	“hot brunette”	2	3	0
zheleznaya distsiplina	“iron discipline”	2	8	0
zheleznyy kharakter	“iron character”	2	0	0
krepkaya druzhba	“strong friendship”	2	1	1
nesterpimaya bol′	“unbearable pain”	1	4	0
ozhestochennyy boy	“fierce battle”	11	2	0
ostraya kritika	“sharp criticism”	1	2	1
ostraya nuzhda	“urgent need”	0	0	0
polnaya svoboda	“complete freedom”	22	13	2
polnaya tishina	“complete silence”	9	21	0
tverdaya uverennost′	“firm confidence”	6	4	0
tyazhelaya bolezn′	“serious illness”	11	9	2

4.2 Textual and syntactic characteristics

Based on the main corpus of the RNC and its textual annotation, it was found that the selected collocations are more characteristic of journalistic texts (compared to fiction). The use of the collocations prevails in the position of the end of the clause. Obviously, it is impossible to use the considered units in plural since abstract nouns cannot be counted, so most examples were found in the singular form. It can also be noted that examples of collocations are more typical for texts written by men.

5 Conclusion

The analyzed collocations are characterized by low occurrences in the corpus. It can be assumed that, on the one hand, dictionary collocations are rare linguistic phenomena, and on the other hand, dictionaries themselves are not an ideal source of data compared with corpora.

The results of this and future work in this direction are essential for developing applications related to speech processing. Creating a full-fledged descrip-

Table 2: Results for the dictionary index 2.

Collocation		Freq (SCR)	Freq (RNC)	Dist(SCR)
bezmernaya glubina	“immeasurable depth”	0	0	0
bezumnaya otvetstvennost′	“terrible responsibility”	0	0	0
bol′shoy poklonnik	“big fan”	7	8	1
vysokiy spros	“high demand”	1	2	0
gromadnaya bystrota	“tremendous speed”	0	0	0
dlinnaya ochered′	“long queue”	6	11	1
doskonal′nyy analiz	“thorough analysis”	0	0	0
isklyuchitel′naya vezhlivost′	“exceptional politeness”	0	1	0
kolossal′naya stoimost′	“colossal cost”	0	0	0
nastoychivaya pros′ba	“insistent request”	1	1	0
nezyblemyy avtoritet	“unshakable authority”	0	0	0
neissyakayemaya vera	“inexhaustible faith”	0	0	0
neistovyy azart	“frantic excitement”	0	0	0
ogromnoye zhelaniye	“great desire”	6	1	0
ogromnyy rost	“huge growth”	5	5	0
ostraya zhalost′	“keen pity”	0	1	0
plamennaya strast′	“fiery passion”	0	0	0
polnoye bezvetriye	“complete calm”	0	1	0
porazitel′naya tishina	“astonishing silence”	1	0	0
reshitel′nyy kharakter	“decisive character”	0	1	0
svezhaya gazeta	“fresh newspaper”	5	1	1
tverdoye obyazatel′stvo	“firm commitment”	0	0	0
tyzhelyy krizis	“severe crisis”	2	0	0
chistoye bezumiye	“pure madness”	2	2	0
chrezmernoye vnimaniye	“excessive attitude”	0	0	1

tive base of Russian oral speech requires a description devoted to stable word combinations. This part is a necessary condition for developing those areas of linguistics and information technologies that take into account a speaker and his (or her) speech behavior.

Acknowledgements. The database was compiled with the support from the Russian Science Foundation (Project No. 19-78-00091). The work on collocation evaluation in spoken corpora was supported by St Petersburg State University, project No. 75254082 “Modeling of Russian megalopolis citizens′ communicative behavior in social, speech and pragmatic aspects using artificial intelligence methods”.

References

1. Borisova, E.: A Word in a Text. A Dictionary of Russian Collocations with English-Russian Dictionary of Keywords [Slovo v tekste. Slovar′ kollokatsiy (ustoychivykh sochetaniy) russkogo yazyka s anglo-russkim slovarem klyuchevykh slov]. Filologiya: Moscow (1995).

2. Corpus of Russian Oral Speech, `http://russpeech.spbu.ru/`. Last accessed 14 Nov 2021.
3. Deribas, V.: Verb–Noun Collocations in Russian [Ustoychivye glagol'no-imennye slovosochetaniya russkogo yazyka]. Russkiy yazyk, Moscow. (1983)
4. The Dictionary of the Russian Language [Slovar' russkogo jazyka v 4 tomakh]. Yevgen'yeva A. P. (ed.-in-chief). Vol. 1–4, 2nd edition, revised and supplemented. Russkij jazyk: Moscow (1981–1984).
5. Khokhlova, M.: Building a Gold Standard for a Russian Collocations Database. In: Proceedings of the XVIII EURALEX International Congress: Lexicography in Global Contexts, pp. 863–-869. Ljubljana (2018)
6. Khokhlova, M.: Collocations in Russian Lexicography and Russian Collocations Database. In: Proceedings of The 12th Language Resources and Evaluation Conference. Marseille, France, pp. 3191–3199. European Language Resources Association (2020)
7. Kustova, G.: Dictionary of Russian Idiomatic Expressions [Slovar' russkoyj idiomatiki. Sochetaniya slov so znacheniyem vysokoy stepeni] (2008), `http://dict.ruslang.ru`. Last accessed 14 Nov 2021.
8. The Large Explanatory Dictionary of the Russian Language [Bol'shoy tolkovyy slovar' russkogo yazyka]. S.A. Kuznetsov (ed.). Norint: St. Petersburg (1998).
9. Mel'čuk, I., Zholkovsky, A.: Explanatory Combinatorial Dictionary of Modern Russian [Tolkovo-kombinatornyy slovar russkogo yazyka]. Vienna. (1984)
10. ORD Speech Corpus ("One Day of Speech"), `https://ord.spbu.ru/`. Last accessed 14 Nov 2021.
11. Oubine, I.: Dictionary of Russian and English Lexical Intensifiers [Slovar' usilitel'nykh slovoso-chetaniy russkogo I angliyskogo yazykov]. Russian Language: Moscow (1987).
12. Project "Night Dream Stories and Other Corpora of Oral Speech", `http://spokencorpora.ru`. Last accessed 14 Nov 2021.
13. Reginina, K., Tjurina, G., Shirokova, L.: Set Expressions of the Russian Language. A Reference Book for Foreign Students [Ustoychivye slovosochetaniya russkogo yazyka: Uchebnoye posobiye dlya studentov-inostrantsev]. Shirokova, L. I. (ed.). Moscow (1980).
14. Russian National Corpus, `http://ruscorpora.ru`. Last accessed 14 Nov 2021.

Website Properties in Relation to the Quality of Text Extracted for Web Corpora

Vít Suchomel[†‡], Jan Kraus[‡]

†Natural Language Processing Centre
Faculty of Informatics, Masaryk University, Brno, Czechia
`xsuchom2@fi.muni.cz`

‡Lexical Computing
Brno, Czechia
`{vit.suchomel,jan.kraus}@sketchengine.eu`

Abstract. In this paper we present our research concerning the relation between two properties of websites and the quality of the text extracted from a website in the context of crawling the web and building large web corpora. A manual classification of text quality of 18 thousand websites from 21 European languages was used to verify our assumption that certain web domain properties can be used to identify potential sources of bad quality content.

The first property is the distance of a web domain from the seed domains in a web crawl. The second property studied in this work is the length of the website name. Although these properties were recommended to help identify good quality websites in our previous work, in this paper we show there is only a small difference between the quality of text-rich web domains with various seed distances or name lengths. This conclusion holds for the post-crawling text processing when starting the web crawl with a large amount of seed domains.

Keywords: Web crawling · Web spam · Text corpus · Text processing

1 Introduction and Motivation

Large web corpora are used in many linguistic, lexicographic and NLP applications. Although the web is a large and easy-to-use source of texts, there is a lot of low quality content. We defined the good and bad content with regards to a linguistic use of text corpora in [1, p. 72]: *A fluent, naturally sounding, consistent text is good, regardless of the purpose of the web page or its links to other pages. The bad content is this: computer generated text, machine translated text, text altered by keyword stuffing or phrase stitching, text altered by replacing words with synonyms using a thesaurus, summaries automatically generated from databases* (*e.g. weather forecast, sport results – all of the same kind very similar*)*, and finally any incoherent text. This is the kind of non-text this work is interested in.*

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 167–175, 2021. © Tribun EU 2021

To get a fluent, naturally sounding and consistent text in the corpus, one should avoid downloading websites providing low quality content and – since that is only partially possible [1, p. 64] – filter out poor quality text from the crawled data as a post-processing procedure. Since the nature of a significant part of non-text is to look like a human-produced text, a human intervention is needed.

We proposed a semi-manual approach consisting in manually checking the largest sources of data and training a non-text classifier, using this data, for the rest of the corpus in [1, p. 85]: *Our assumption in this setup is that all pages in a web domain are either good – consisting of nice human produced text – or bad – i.e. machine generated non-text or other poor quality content. Although this supposition might not hold for all cases and can lead to noisy training data for the classifier, it has two advantages: Much more training samples are obtained and the cost to determine if a web domain tends to provide good text or non-text is not high.*

This paper presents the process of the manual check of text quality of large websites in the corpus in chapter 2.

Furthermore, we were interested in the usefulness of web domain properties for assessing the quality of the text yielded by the site. Some properties are evaluated on-the-fly by web crawler SpiderLing [2] that is used by us to crawl the web. Selected web domain characteristics are described in chapter 3. The relation of these metrics to the website quality is dealt with in chapter 4. This research broadens the evaluation reported in [1, p. 90] to 18 thousand websites from 21 European languages.

2 Checking Website Text Quality in Large Web Corpora

Here follows the procedure of checking website text quality in TenTen web corpora [3] we build for text corpus management system Sketch Engine [4].

The number of websites to be checked is proportionate to the size of the domains in tokens. If a domain contains more than 10 million tokens, a higher priority will be given to such domain. On the other hand, if a domain contains less than 2 million tokens, there will be a lower priority during the checking process and this often creates the threshold, i.e. smaller websites will not be manually checked, since their impact on the corpus quality is marginal.

On average it is possible to manually check about 50 to 70 domains per hour, depending on the familiarity with the language, language script, etc. The size of the language also plays a role. Languages like English, Spanish, German etc. are much more extensive in content (tens of billions of tokens) and that is why a larger number of domains will be manually checked, usually 2,000 to 5,000. For smaller corpora (billions of tokens), the number of websites to check will be usually about 300 to 500.

The first step in web domain checking is to pick the largest domains that make up the majority of the corpus, usually that is at least 50 % of the corpus, depending on the total size of the corpus and language. The second step is to check random concordances of three consecutive sentences from the selected web domains. This concordance usually consists of 50–70 lines.

The third step involves a manual checking of the random concordances. One of the most important things when determining whether to keep a specific domain in our corpora is the genuineness of the texts. After the web domains are downloaded, there might be a certain percentage of spam and other texts of lesser quality impacting the corpus quality and such texts must be removed from the corpus. During this phase of checking, each domain is either labelled as *ok* or *bad*. The domains labeled as bad contain either spam (generated text without any meaning) or machine translated texts, which might be difficult to spot in languages we do not know in depth. In such cases the website source code, domain name or the live website will usually give clues.

Apart from this, there might be other criteria for keeping web domains in corpora. If a certain domain contains a large amount of lists, square brackets, angle tag brackets or other non-text elements, these domains will be tagged as bad and thus removed from the corpus. Sometimes this decision will depend on the language and corpus size. Especially if the corpus is rather small, for instance no more than one billion words, such texts might be preserved for the sake of having some linguistic data and meeting the first condition of the text being a spam or not will suffice.

After this phase of checking is completed, there might be other ways to identify the bad content. Since some of the bad domains were already identified in the previous step, we can use some of the words present in bad domains to run a concordance search to find other bad domains. This step usually works for spam. If spam contains words like „porn“, „xxx“, „viagra“ etc., other bad domains might be identified this way.

3 Selected Web Domain Properties

The data is obtained from the internet through crawling – starting from seed URLs (or domains), downloading web pages (or other documents) and following links found in these pages. We selected two web domain properties evaluated on-the-fly by web crawler SpiderLing [2]: The distance of a web domain from the seed domains and the length of the website name. In addition to text yield ratio, these characteristics are used by the crawler to determine which sources to focus on.

Assuming the web is an oriented graph with web pages being the nodes and links being the vertices, the lowest graph distance from the seed (initial) web pages to a web page in a website is the *domain distance* of the web domain. The domain distance is measured by the crawler. The distance of a domain is heavily dependent on the seed domains and it can vary for different runs or settings of the crawler. The crawler is set to download more often from websites with a short distance.[1]

The *hostname length* is the length of the name of the website, i.e. the hostname character count. The crawler is set to ignore sites with hostname length greater than 40 and to download more often from websites with a short name.[1]

4 The Quality of Text in Relation to Website Properties

The quality of text in web domains human-labelled by *ok* or *bad* is shown in relation to hostname length and domain distance in the following tables and charts. In our corpus building projects, the crawling is usually started from all URLs known to us in the target language, including the previous versions of the corpus. Thus not only trustworthy domains (such as news sites, government webs and site whitelists [5]) are in distance 0. That means we care less for avoiding bad sites and identify them in the post-processing phase to discover as many links to good parts of the web (hopefully) as possible.

Note this is an evaluation of the largest text sources in a particular language (i.e. from a website containing documents in the language) that were downloaded by the crawler already giving priority to domains with a short distance or a short hostname.

The text quality by domain distance for 18 thousand websites from 21 European languages is shown in Fig. 1. The same data is evaluated with regards to hostname length in Fig. 2. A zero distance or a very short name is somewhat indicating a good content. Based on this findings, we do not recommend using the domain distance in decisions about text quality in post-processing when the crawler started with all URLs available rather than a trustworthy seeds. That is also the main difference from conclusions based on the chart in [1, p. 90].

A detailed breakdown of the counts of good and bad domains grouped by the domain distance or the hostname length can be found in Table 1.

The detailed figures for selected separate languages are presented in Table 2 for Czech, in Table 3 for Slovene, in Table 4 for Polish, in Table 5 for German and in Table 6 for Latvian.

---

[1] This measure has an impact just for crawls with a large number of domains in the download queue, mainly the English web, since all domains are scheduled for download anyway in case there is less domains to choose from.

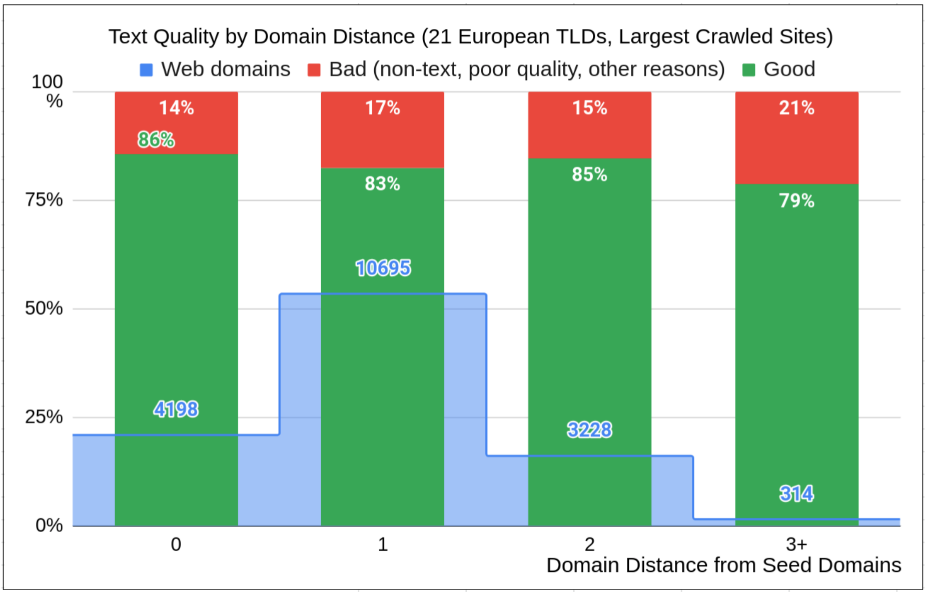


Fig. 1: Text quality by domain distance, all data from this report together. The proportion of good and bad domains is shown in green and red, respectively. The number of web domains in each band is displayed by the blue stepped chart.

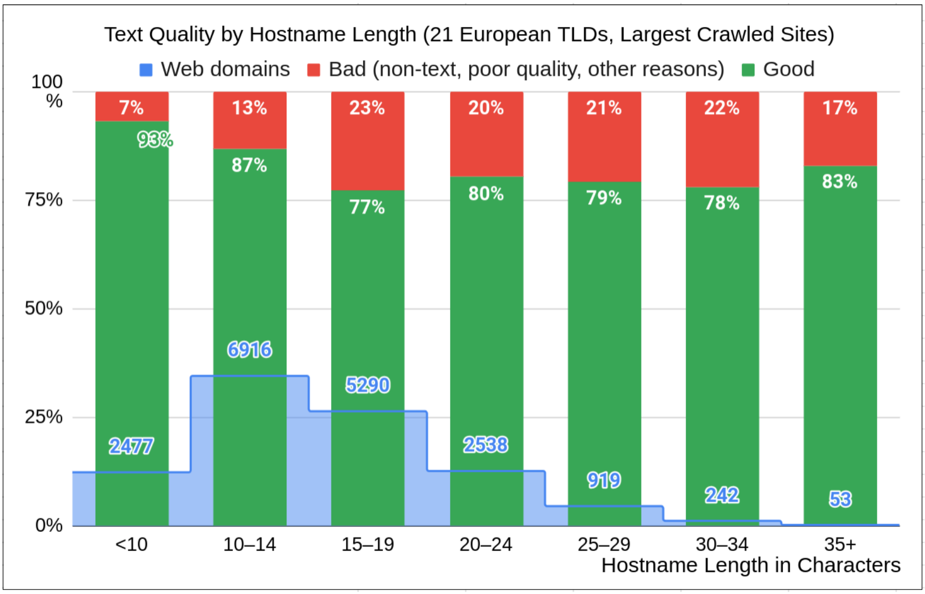


Fig. 2: Text quality by hostname length, all data from this report together. The proportion of good and bad domains is shown in green and red, respectively. The number of web domains in each band is displayed by the blue stepped chart.

Table 1: Domain count analysis for all data in Fig. 1 and Fig. 2.

21 European languages	**domains**	**ok**	**bad**
domains	18529	83%	16%
median distance		1	1
median name length		14	16

distance	**domains**	**ok**	**bad**
0	4239	85%	14%
1	10738	82%	17%
2	3238	84%	15%
3+	314	79%	21%

name length	**domains**	**ok**	**bad**
<10	2482	93%	7%
10–14	6953	86%	13%
15–19	5323	77%	23%
20–24	2552	80%	20%
25–29	924	79%	21%
30–34	242	78%	22%
35+	53	83%	17%

Table 2: Domain count analysis for a 2019 crawl of Czech. The domain distance is unrelated to data quality. The hostname length is somewhat related to data quality.

Czech Web 2019	**domains**	**ok**	**bad**
domains	878	91%	9%
median distance		2	2
median name length		12	14

distance	**domains**	**ok**	**bad**
0	244	94%	6%
1	154	84%	16%
2	426	92%	8%
3+	54	91%	9%

name length	**domains**	**ok**	**bad**
<10	181	96%	4%
10–14	396	90%	10%
15–19	238	90%	10%
20–24	56	89%	11%
25–29	5	60%	40%
30–34	2	100%	0%

Table 3: Domain count analysis for a 2020 crawl of Slovene. The measures are almost unrelated to data quality here.

Slovene Web 2020	**domains**	**ok**	**bad**
domains	250	91%	9%
median distance		1	1
median name length		13	14

distance	**domains**	**ok**	**bad**
0	65	95%	5%
1	155	93%	7%
2	29	72%	28%
3+	1	100%	0%

name length	**domains**	**ok**	**bad**
<10	40	95%	5%
10–14	107	91%	9%
15–19	77	90%	10%
20–24	23	91%	9%
25–29	2	100%	0%
30–34	0		
35+	1	100%	0%

Table 4: Domain count analysis for a 2019 crawl of Polish. The measures are unrelated to data quality here.

Polish Web 2019	**domains**	**ok**	**bad**
domains	762	91%	9%
median distance		1	0
median name length		14	13

distance	**domains**	**ok**	**bad**
0	299	87%	13%
1	431	94%	6%
2	31	94%	6%
3+	1	100%	0%

name length	**domains**	**ok**	**bad**
<10	124	90%	10%
10–14	318	92%	8%
15–19	223	91%	9%
20–24	79	94%	6%
25–29	17	88%	12%
30–34	1	0%	100%

Table 5: Domain count analysis for a 2020 crawl of German. The measures are unrelated to data quality here.

German Web 2020	**domains**	**ok**	**bad**
domains	2398	94%	4%
median distance		1	1
median name length		14	15

distance	**domains**	**ok**	**bad**
0	592	89%	7%
1	1614	96%	3%
2	189	97%	3%
3+	3	100%	0%

name length	**domains**	**ok**	**bad**
<10	326	97%	2%
10–14	893	94%	5%
15–19	753	92%	6%
20–24	299	97%	2%
25–29	100	95%	2%
30–34	26	92%	8%
35+	1	100%	0%

Table 6: Domain count analysis for a 2019 crawl of Latvian. The domain distance is rather negatively related to data quality, it seems like the crawler found a better content then was yielded by the initial sites. The hostname length is related to data quality well.

Latvian Web 2021	**domains**	**ok**	**bad**
domains	453	46%	54%
median distance		1	0
median name length		12	18

distance	**domains**	**ok**	**bad**
0	198	34%	66%
1	235	56%	44%
2	17	53%	47%
3+	3	0%	100%

name length	**domains**	**ok**	**bad**
<10	57	91%	9%
10–14	125	85%	15%
15–19	254	17%	83%
20–24	14	36%	64%
25–29	3	33%	67%

5 Conclusions

In this paper we have described the website checking part of the process of extraction and cleaning text from the Internet for building large web corpora in Sketch Engine. The relations of web domain seed distance and hostname length to the quality of the website content were studied using 18 thousand websites from 21 European languages.

We found there is none or a small difference between the content quality of text-rich web domains and the domain distance. The host name length is somewhat related to the domain text quality. Both relations depend on the particular crawl setup.

Although the studied website properties may be helpful for the crawler's scheduler to decide which small domains to visit more frequently, they are not related much to the text quality of the largest websites when starting the web crawl with a large amount of seed domains.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015.

References

1. Suchomel, V.: Better Web Corpora For Corpus Linguistics And NLP. PhD thesis, Masaryk University (2020)
2. Suchomel, V., Pomikálek, J.: Efficient web crawling for large text corpora. In: Proceedings of the seventh Web as Corpus Workshop (WAC7). (2012) 39–43
3. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. International Conference on Corpus Linguistics, Lancaster (2013)
4. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: ten years on. Lexicography **1** (2014)
5. Baisa, V., Suchomel, V.: Skell: Web interface for english language learning. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2014, Brno (2014) 63–70

Development of HAMOD: a High Agreement Multi-lingual Outlier Detection dataset

Miloš Jakubíček[†], Emma Romani[‡], Pavel Rychlý[†], Ondřej Herman[†]

†Natural Language Processing Centre
Faculty of Informatics, Masaryk University, Brno, Czechia
`{jak,pary,xherman1}@fi.muni.cz`

†Lexical Computing
Brno, Czechia
`{milos.jakubicek,pavel.rychly,ondrej.herman}@sketchengine.eu`

‡Università degli studi di Pavia
Pavia, Italy
`emma.romani01@universitadipavia.it`

Abstract.
In this paper we describe further development of a High Agreement Multi-lingual Outlier Detection dataset (HAMOD) outlier that is used for the purpose of evaluation of automatic distributional thesauri. We briefly introduce the task and methodological motivation for developing such a dataset, then we present the current status of the dataset and related tools as well as results measured on the dataset so far (both in terms of agreement rates and thesauri eveluation). Finally we discuss future developments of HAMOD.

Keywords: HAMOD · Distributional thesaurus · Outlier detection · Word embeddings · Sketch Engine

1 Introduction and motivation

This paper presents new developments of the HAMOD dataset. HAMOD stands for an acronym of *High Agreement Multi-lingual Outlier Detection*, a dataset for exercising the outlier detection task that aims at high inter-annotator agreement. Outlier detection is a task where a human or machine is presented with a set of words (in our case 9), out of which one is a so called *outlier*: a word that "doesn't fit" to the others.

In [1] it was argued that outlier detection is (unlike the intrinsic evaluation based on similarity judgements) a reliable method for evaluating automatic distributional thesauri. A distributional thesaurus is generally a mapping of pairs of words to a numeric similarity score (or conversely, a dissimilarity score, i.e. a distance) yielding in the first place a list of most similar words for a given word. There are several methods for calculating a distributional thesaurus, such as using word sketches in Sketch Engine [2] or using a vector space model

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 177–183, 2021. © Tribun EU 2021

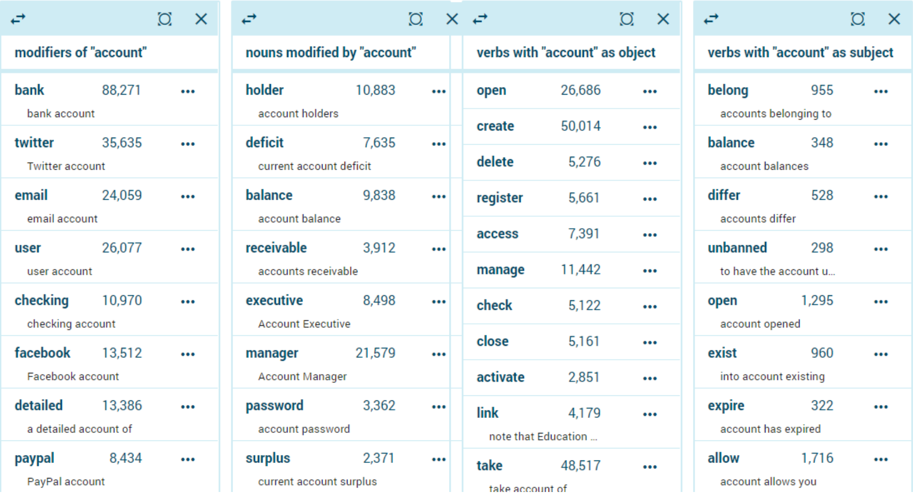
(word embeddings) (see e.g. [3]). The real difficulty for any comparison and further development of these methods is that a reliable evaluation methodology is currently missing: a directly intrinsic evaluation suffers from extremely low inter-annotator agreement. For this reason we started developing HAMOD in 2019 and continuously expand the dataset both in terms of number of languages and number of exercises.

In further text we describe the dataset itself, thesauri that we used for evaluation so far and our plans for further development.

2 Sketch Engine and the word sketch-based thesaurus

Sketch Engine [4] is a leading text corpus management system which as of 2021 includes several hundreds of preloaded corpora as well as corpus-building functionalities available for regular end users. The preloaded corpora typically come from the web and aim at targeting multi-billion size. In 2010, Sketch Engine started the so-called TenTen series of web corpora [5], aiming at building a corpus of ten billion words (10^{10}, thus "TenTen") for as many languages as possible.

A word sketch is a short summary of a word's collocational behaviour from the perspective of individual grammatical relations (noun's modifier, verb's subject etc.), as can be seen from the example given in Figure 1.



modifiers of "account"		
bank	88,271	bank account
twitter	35,635	Twitter account
email	24,059	email account
user	26,077	user account
checking	10,970	checking account
facebook	13,512	Facebook account
detailed	13,386	a detailed account of
paypal	8,434	PayPal account

nouns modified by "account"		
holder	10,883	account holders
deficit	7,635	current account deficit
balance	9,838	account balance
receivable	3,912	accounts receivable
executive	8,498	Account Executive
manager	21,579	Account Manager
password	3,362	account password
surplus	2,371	current account surplus

verbs with "account" as object		
open	26,686	
create	50,014	
delete	5,276	
register	5,661	
access	7,391	
manage	11,442	
check	5,122	
close	5,161	
activate	2,851	
link	4,179	note that Education ...
take	48,517	take account of

verbs with "account" as subject		
belong	955	accounts belonging to
balance	348	account balances
differ	528	accounts differ
unbanned	298	to have the account u...
open	1,295	account opened
exist	960	into account existing
expire	322	account has expired
allow	1,716	account allows you

Fig. 1: An example of a word sketch for the English noun *account*.

Each word sketch item is a triple consisting of the headword, the grammatical relation and the collocate. As such a word sketch is basically a dependency syntax graph, calculated using a hybrid rule-based and statistical approach. The backbone word for computing word sketches represents a hand-written word sketch grammar, which selects collocation candidates using the corpus query language (CQL, [6]).

A sketch grammar typically makes heavy use of regular expressions over morphological annotation of the corpus to select syntactically viable collocation candidates. These candidates are subsequently subject to statistical scoring using a word association score. LogDice is used as the association metric in Sketch Engine as it was proven to be scalable across corpora of different sizes and produces scores comparable across corpora too [7].

Word sketches make it possible to automatically derive a distributional thesaurus by calculating similarity of word sketch contexts: for each word, we look at which other words share most collocates (in the same grammatical relations).

To compute a similarity score between word w_1 and word w_2, we compare w_1 and w_2's word sketches in this way:

- find all the overlaps, i. e. where w_1 and w_2 share a collocation in the same grammatical relation, e. g.: $(beer/wine, OBJECT\_OF, drink)$, where the association score > 0,
- let ws_{w_1} and ws_{w_2} be the set of all word sketch triples $(headword, relation, collocation)$ for w_1 and w_2, respectively, where the association score > 0,
- let $ctx(w_1) = \{(r,c)|(w_1,r,c) \in ws_{w_1}\}$,
- let AS_i be the association score of a word sketch triple (logDice),
- then the distance between w_1 and w_2 is computed as:

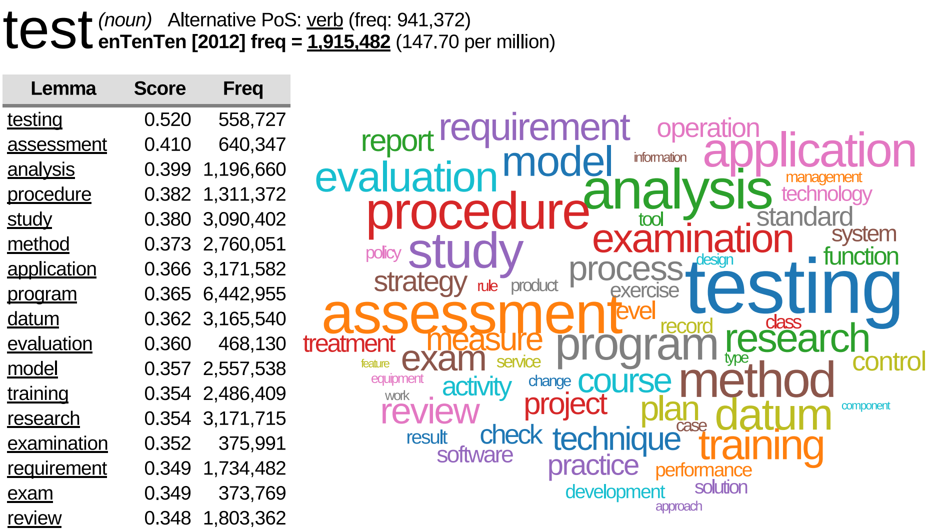
$$Dist(w_1,w_2) = \frac{\sum\limits_{(r,c)\in ctx(w_1)\cap ctx(w_2)} AS_{(w_1,r,c)} + AS_{(w_2,r,c)} - \frac{\left(AS_{(w_1,r,c)} - AS_{(w_2,r,c)}\right)^2}{50}}{\sum_{i\in ws_1} AS_i + \sum_{i\in ws_2} AS_i}$$

The term $(AS_i - AS_j)^2/50$ is subtracted in order to give less weight to shared triples, where the triple is far more salient with w_1 than w_2 or vice versa. We find that this contributes to more readily interpretable results, where words of similar frequency are more often identified as near neighbours of each other.

A thesaurus screenshot from Sketch Engine can be found in Figure 2.

3 Thesaurus built from word embeddings

Another method, or rather a whole paradigm, that can be used for deriving an distributional thesaurus, is based on calculating a vector representation for each word in a corpus (so called word embedding) and using the distances between individual word vectors as a measure of words' (dis)similarity. For our experiments we used FastText [8] and Word2vec [3] to calculate word embeddings based on corpora available in Sketch Engine [9].



test *(noun)* Alternative PoS: verb (freq: 941,372)
enTenTen [2012] freq = 1,915,482 (147.70 per million)

Lemma	Score	Freq
testing	0.520	558,727
assessment	0.410	640,347
analysis	0.399	1,196,660
procedure	0.382	1,311,372
study	0.380	3,090,402
method	0.373	2,760,051
application	0.366	3,171,582
program	0.365	6,442,955
datum	0.362	3,165,540
evaluation	0.360	468,130
model	0.357	2,557,538
training	0.354	2,486,409
research	0.354	3,171,715
examination	0.352	375,991
requirement	0.349	1,734,482
exam	0.349	373,769
review	0.348	1,803,362

report
requirement
operation
model
information
application
evaluation
management
analysis
technology
procedure
tool
standard
system
examination
policy
study
function
design
strategy
rule
product
process
exercise
testing
assessment
level
record
class
treatment
measure
research
program
type
feature
exam
service
control
equipment
activity
change
course
method
work
review
project
plan
datum
component
case
result
check
technique
training
software
practice
performance
development
solution
approach

Fig. 2: An example of the thesaurus for the English noun *test*.

Unlike the corpora used for the word-sketch based thesaurus, corpora used for training word embeddings do not need to be part-of-speech tagged or lemmatized, on the other hand our preliminary observations showed that much larger datasets are required. This observation is to be expected and represents a typical data richness vs. data size trade-off.

4 Building HAMOD

In 2019 we started building HAMOD, initially on a set of three languages (English, Czech and Slovak). Currently, four other languages were added (Estonian, French, German and Italian) and we plan to expand the dataset further on. New languages are added by translating from English but where the translation results into ambiguities in the target language, we adjust the exercise set accordingly. Thus the dataset is not strictly a parallel one but a comparable one. Each exercise set of HAMOD contains 8 inliers, i.e. words that are part of a semantic category or together define a topic an, and 8 outliers. In each exercise all inliers and one outlier is presented, thus we have 8 exercises available for each such exercise set.

Since key aspect of HAMOD is the high agreement, we developed a simple web interface for exercising the outlier detection tasks by human evaluators. We aim at having at least 10 independent evaluations for each exercise and each human evaluator should be presented with an exercise set only once (i.e. never multiple times with different outliers where we could reuse the information from previous run), therefore we need 80 evaluators at minimum for each language. After completing the whole exercise, we present the evaluator with an overall success score, but do not disclose individual discrepancies.

A screenshot from the web inteface used for evaluation is provided in Figure 3. In each turn of the exercise, evaluators select the outlier, or may skip the turn if they are unsure. Currently HAMOD contains 38 complete exercise sets and the target size for all languages is 100.

5 Evaluation

Initial evaluation of the inter-annotator agreement for Czech and Estonian shows very promising results as it exceeds 90 % of absolute raw agreement (chance-correction does not play a big role: with 10 annotators and 8 options chance agreement is $\frac{1}{8}^{10} < 10^{-10}$). Detailed agreement figures for both languages are provided in Table 1.

Table 1: Inter-annotator agreement for languages included in HAMOD. A success run means an excercise where all sets where correctly fulfilled by an evaluator.

Language	**Success runs**	**All runs**	**Agreement**
Czech	2,082	2,150	0.97
Estonian	3,285	3,525	0.93

Evaluation of two distributional thesauri by means of overall accuracy (where the outlier was correctly identified) and outlier position percentage (OPP, average percentage of the right answer) is provided in Table 2. We used the czTenTen12, deTenTen13, enTenTen13, frTenTen12, itTenTen16, skTenTen11 [5] and EstonianNC 2017 [10] corpora available in Sketch Engine. For a detailed description of the evaluation, see [1].

The evaluation of the thesauri is clearly just a starting point but it already shows that none of the variants (thesaurus based on word sketches and thesaurus based on word embeddings) outperforms the other one for all languages.

6 Conclusions and future development

In this paper we have described recent developments of the HAMOD dataset. We argued why such a dataset is necessary for further development, evaluation and comparison of distributional thesauri and we have discussed the current status of the dataset. We plan to further expand the dataset to reach 100 exercises sets and cover more languages (EU languages in the first place) while continuously monitoring the inter-annotator agreement and adjusting the dataset accordingly to maintain high agreement. So far the discriminative power of the dataset (i.e. its ability to discover differences between individual thesaurus types) is maintained as well but we are aware of the fact that at

Table 2: Comparison of a Sketch Engine-based and word-embeddings-based thesaurus on the HAMOD dataset. Dataset size means number of exercises (outlier detection exercise sets) that were evaluated.

Corpus	**Corpus size**	**Dataset size**	**SkE Acc**	**SkE OPP**	**Word2Vec Acc**	**Word2vec OPP**
czTenTen12	5G	232	0.573	0.898	0.655	0.871
enTenTen13	22G	296	0.456	0.847	0.655	0.873
EstonianNC 2017	1.3G	296	0.564	0.832	0.547	0.784
deTenTen13	19G	232	0.349	0.798	0.323	0.764
frTenTen12	6.8G	232	0.276	0.744	0.427	0.768
skTenTen11	0.6G	296	0.389	0.777	0.591	0.851
itTenTen16	5.8G	296	0.453	0.856	0.581	0.869

some point of further development of the thesauri the dataset might need to be revisited if it looses its discriminative power, i.e. if it would be a task too easy for the computer. When finished the dataset will become available under a permissible Creative Commons licence in a public repository.

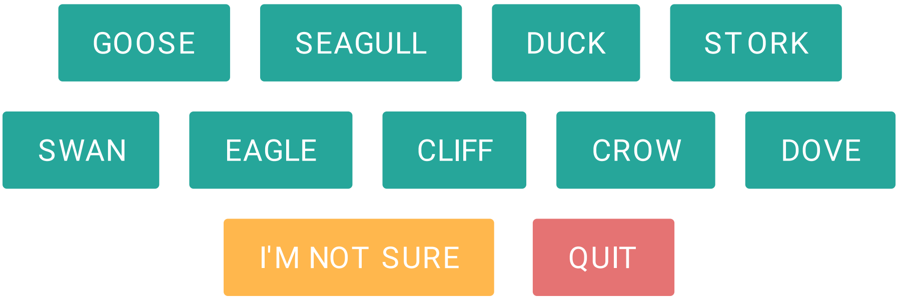


Fig. 3: A sample outlier detection exercise generated for English.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015.

References

1. Rychlý, P.: Evaluation of czech distributional thesauri. In: RASLAN 2019 Proceedings of Recent Advances in Slavonic Natural Language Processing. (2019) 137–142

2. Rychlý, P., Kilgarriff, A.: An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, Association for Computational Linguistics (2007) 41–44
3. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pretraining distributed word representations. arXiv preprint arXiv:1712.09405 (2017)
4. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: ten years on. Lexicography **1** (2014)
5. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. International Conference on Corpus Linguistics, Lancaster (2013)
6. Jakubíček, M., Rychlý, P., Kilgarriff, A., McCarthy, D.: Fast syntactic searching in very large corpora for many languages. In: PACLIC 24 Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, Tokyo (2010) 741–747
7. Rychlý, P.: A lexicographer-friendly association score. RASLAN 2008 Proceedings of Recent Advances in Slavonic Natural Language Processing (2008) 6–9
8. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5** (2017) 135–146
9. Herman, O.: Precomputed word embeddings for 15+ languages. RASLAN 2021 Proceedings of Recent Advances in Slavonic Natural Language Processing (2021)
10. Koppel, K.: Näitelausete korpuspõhine automaattuvastus eesti keele õppesõnastikele. Tartu Ülikooli Kirjastus (2020)

Subject Index

activity 71, 83
automatic alignment 115

collocations 161
coordination 125
corpus 19, 41, 115, 135, 167
– parallel 115
– spoken 161
Czech 3, 61, 107, 135

debiasing 91
dictionary 3, 141, 161
dictionary writing system 3
domain
– adaptation 91
– extrapolation 91
– robustness 91

education 91
English 141
evaluation 61, 141
explications 49

FCA 49

generalization 91
ground truth dictionary 141

HAMOD 177
health records 151

image super-resolution 11, 29

language identification 29
layout analysis 29
lemmatization 135

medieval texts 11, 29
MongoDB database 3
morphological guesser 135
morphology 135

named entity recognition 151
neural language models 91

online risks 19
ontology 83
optical character recognition 11, 29
outlier detection 177

Polish 151
proofreading 107
punctuation 107

question answering 71
– answer context 61
– answer selection 61

regular expressions 107
Russian 161

sign language 3
Sketch Engine 41, 177
Slavic languages 151
Slovak 141
spam 167
supportive interaction 19
syntactic analysis 125

text classification 19
text processing 167
thesaurus 177
transformer models 19
– BERT 91
Transparent Intensional Logic 71

verb-valency frames 83
VerbaLex 125

web crawling 167
web spam 167
word embeddings 41, 177
– cross-lingual 141

XML database 3

zeugma 125

Author Index

Anetta, K. 151
Arslan, M. 151

Bankovič, M. 11

Číhalová, M. 83

Denisová, M. 141
Duží, M. 71

Herman, O. 41, 177
Horák, A. 29, 61

Jakubíček, M. 177

Khokhlova, M. 161
Kovář, V. 135
Kraus, J. 167

Lebedíková, M. 19

Medková, H. 125
Medveď, M. 61

Mrkývka, V. 107

Novotný, V. 11, 29

Plhák, J. 19

Rambousek, A. 3
Romani, E. 177
Rychlý, P. 135, 141, 177

Sabol, R. 61
Seidlová, K. 29
Signoroni, E. 115
Šmahel, D. 19
Sojka, P. 11, 91
Sotolář, O. 19
Štefánik, M. 91
Suchomel, V. 167

Tkaczyk, M. 19

Vrabcová, T. 29

RASLAN 2021

Fifteenth Workshop on Recent Advances in Slavonic Natural Language Processing

Editors: Aleš Horák, Pavel Rychlý, Adam Rambousek
Typesetting: Adam Rambousek
Cover design: Petr Sojka

Published and printed by Tribun EU
Cejl 892/32, 602 00 Brno, Czech Republic

First edition at Tribun EU
Brno 2021

ISBN 978-80-263-1670-1
ISSN 2336-4289