RASLAN 2020
Recent Advances in Slavonic Natural Language Processing

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

RASLAN 2020

Recent Advances in Slavonic Natural Language Processing

Fourteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2020
Brno (on-line), Czech Republic, December 8–10, 2020
Proceedings

Tribun EU
2020

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

ISBN 978-80-263-1600-8
ISSN 2336-4289

Preface

This volume contains the Proceedings of the Fourteenth Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2020) originally planned to be held on December 6th–8th 2020 in Karlova Studánka, Sporthotel Kurzovní, Jeseníky, Czech Republic. Due to the ongoing pandemic of COVID-19 and related governmental restrictions, RASLAN 2020 was held as an on-line event.

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 2020 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 13 papers were accepted, contributed altogether by 23 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 Marie Stará. 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 2020 Karel Pala

Table of Contents

I NLP Applications

When Tesseract Does It Alone 3
Vít Novotný

Data Mining from Free-Text Health Records 13
Krištof Anetta

Efficient Management and Optimization of Very Large Machine Learning Dataset for Question Answering 23
Marek Medveď, Radoslav Sabol, and Aleš Horák

II Semantics and Language Modelling

Towards Useful Word Embeddings 37
Vít Novotný, Michal Štefánik, Dávid Lupták, and Petr Sojka

Using FCA for Seeking Relevant Information Sources 47
Marek Menšík, Adam Albert, and Vojtěch Patschka

The Art of Reproducible Machine Learning 55
Vít Novotný

III Morphology and Syntax

Multilingual Recognition of Temporal Expressions 67
Michal Starý, Zuzana Nevěřilová, and Jakub Valčík

Automatic Detection of Zeugma 79
Helena Medková

Cthulhu Hails from Wales 87
Vít Novotný and Marie Stará

IV Text Corpora

RapCor, Francophone Rap Songs Text Corpus 95
Alena Podhorná-Polická

Semantic Analysis of Russian Prepositional Constructions 103
Victor Zakharov, Kirill Boyarsky, Anastasia Golovina, and Anastasia Kozlova

Removing Spam from Web Corpora Through Supervised Learning and Semi-manual Classification of Web Sites 113
Vít Suchomel

Evaluating Russian Adj-Noun Word Sketches against Dictionaries: a Case Study 125
Maria Khokhlova

Subject Index 133

Author Index 135

Part I

NLP Applications

When Tesseract Does It Alone
Optical Character Recognition of Medieval Texts

Vít Novotný

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

Abstract. Optical character recognition of scanned images for contemporary printed texts is widely considered a solved problem. However, the optical character recognition of early printed books and reprints of Medieval texts remains an open challenge. In our work, we present a dataset of 19th and 20th century letterpress reprints of documents from the Hussite era (1419–1436) and perform a quantitative and qualitative evaluation of speed and accuracy on six existing OCR algorithms. We conclude that the Tesseract family of OCR algoritms is the fastest and the most accurate on our dataset, and we suggest improvements to our dataset.

Keywords: Optical character recognition, OCR, Historical 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 at the Czech Medieval Sources online (CMS online) web site,[1] accurate optical character recognition (OCR) algorithms are required to extract searchable text from the scanned images.

In this paper, we compare the speed and accuracy of six OCR algorithms on the CMS online dataset both quantitatively and quantitatively. In Section 2, we describe the OCR algotithms. In Section 3, we describe our dataset, how it was pre-processed and used in our quantitative and qualitative evaluation. 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

Optical character recognition makes it possible to convert scanned images to digital text. For the AHISTO project, an OCR algorithm should:

[1] https://sources.cms.flu.cas.cz/

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 3–12, 2020.

1. support different European languages, mainly Czech, German, and Latin,
2. detect language of the text to enhance the searchability of our database, and
3. use a standard output format to prevent a vendor lock-in.

In this section, we will describe and discuss the OCR algorithms we considered.

2.1 Google Cloud Vision AI

Google Cloud Vision AI is a paid OCR service made available by Google in 2015.[2]

Google Vision AI supports Czech, German, and Latin, among other languages,[3] and detects language at the level of individual letters. Regrettably, only a non-standard JSON output format is supported, leading to a vendor lock-in.

Google Cloud Vision AI provides two features: `DOCUMENT_TEXT_DETECTION` and `TEXT_DETECTION`. `DOCUMENT_TEXT_DETECTION` performs the OCR of printed documents, whereas `TEXT_DETECTION` solves the more general task of detecting text in arbitrary images, such as camera images of traffic signs. In our experiments, we used the `DOCUMENT_TEXT_DETECTION` feature.

2.2 Tesseract

Tesseract [6] has been developed by Hewlett-Packard in the 1980s and released under a free open-source license in 2005. Since 2006, the development of the project has been funded by Google, and it is presumed that parts of Tesseract are also used in Google Cloud Vision AI.

Initially, Tesseract only supported English. Since version 2, Tesseract has also supported Western languages, including German. Since version 3, Tesseract has also supported Czech, [7], detected language at the level of words, and added support for the standard HOCR XML output format. [1] Since version 3.04, Tesseract has also supported Latin. Since version 4, Tesseract has supported a new LSTM-based OCR engine,[4] which is regrettably not GPU-accelerated.

In our experiments, we used Tesseract 4.00 in three configurations:

1. `--oem 0`, which uses the non-LSTM OCR engine (f.k.a. Tesseract 3),
2. `--oem 1`, which uses the LSTM OCR engine (f.k.a. Tesseract 4), and
3. `--oem 2`, which ensembles the two OCR engines (f.k.a. Tesseract 3 + 4).

For all configurations, we used the `--psm 3` page segmentation mode, the Czech, German, and Latin (`-l ces+deu+lat`) language models, and the medium-size pre-trained models,[5] which, compared to the best models,[6] support the non-LSTM OCR engine of Tesseract 3.

[2] https://cloud.google.com/vision/docs/release-notes
[3] https://cloud.google.com/vision/docs/languages#supported-langs
[4] https://tesseract-ocr.github.io/tessdoc/NeuralNetsInTesseract4.00
[5] https://github.com/tesseract-ocr/tessdata.git
[6] https://github.com/tesseract-ocr/tessdata_best.git

2.3 OCR-D

OCR-D [2] is a free open-source project that has been funded by the German Research Foundation and developed by the Berlin-Brandeburg Academy of Sciences, the Herzog August Bibliothek, the Berlin State Library, and the Karlsruhe Institute of Technology. The main goal of the project is to digitize the German cultural heritage 16th–19th century in connection with the VD16, VD17, and VD18 cataloging and archival projects. In February 2020, the project has entered phase 3 and OCR-D is now being deployed to intent organizations.

Unlike Google Cloud Vision AI and Tesseract, which provide fully-automated OCR workflows, OCR-D's workflows are fully-configurable[7] and include image enhancement, binarization, cropping, denoising, deskewing, dewarping, segmentation, clipping, line OCR, text alignment, and post-correction. Another distinguishing feature of OCR-D is the OCR4all web frontend,[8] which enables semi-automatic OCR with human input.

As a part of the line OCR workflow step, OCR-D supports the LSTM-based and GPU-accelerated Calamari engine,[9] which has achieved the state-of-the-art performance on historical texts typeset in Fraktur. [9,4] With Calamari, OCR-D regrettably does not detect the language of the text, but it supports the standard HOCR XML output format [1] like Tesseract.

In our experiments, we used the recommended workflow for OCR-D,[10] which includes Calamari as the line OCR engine. Since Calamari is GPU-accelerated, we tested OCR-D both with and without a GPU to see the difference in speed.

3 Methods

In this section, we will describe the CMS online dataset, how it was pre-processed and then used to evaluate the OCR algorithms discussed in the previous section.

3.1 Data Preprocessing

In the CMS online dataset, we received 302,909 low-resolution and 168,113 high-resolution scanned images of letterpress reprints from the 19th and 20th centuries. For all low-resolution images, we received Google Cloud Vision AI OCR outputs, although it is unknown if these were produced by the `TEXT_DETECTION` feature, or the more appropriate `DOCUMENT_TEXT_DETECTION` feature, and if they were produced using the low-resolution or the high-resolution images.

Although the high-resolution images have an estimated average density of 414 DPI and are suitable for OCR, the low-resolution images have an estimated average density of only 145 DPI. Therefore, we had to design an algorithm that would link the low-resolution images to the matching high-resolution images in

[7] https://ocr-d.de/en/workflows
[8] https://www.uni-wuerzburg.de/en/zpd/ocr4all/
[9] https://github.com/Calamari-OCR/calamari
[10] https://ocr-d.de/en/workflows#best-results-for-selected-pages

order to produce a test dataset containing high-resolution images together with corresponding Google Cloud Vision AI OCR outputs as ground truth.

In designing the algorithm, we assumed that there was no other difference between matching low-resolution and high-resolution images other than downscaling. Additionally, we manually inspected the low-resolution and high-resolution images to find a subset of 187,267 (62%) low-resolution images guaranteed to cover all the books from which the high-resolution images originated.

A pseudocode of our linking algorithm is given in Algorithm 1. Using the algorithm, we linked 65,348 (39%) high-resolution scanned images with Google Vision AI ground truth OCR outputs, which served as our test dataset.

Algorithm 1: Linking low-resolution and high-resolution images.

```
Result: Linked low-resolution and high-resolution images.
Preprocess all images by rescaling them to 512 × 512 px and binarizing;
Index the preprocessed high-resolution images in a vector database;
foreach preprocessed low-resolution image do
    Retrieve the ground truth Google Vision AI output;
    Retrieve the 100 preprocessed high-resolution images nearest to the
     preprocessed low-resolution image by the Hamming distance;
    foreach neighboring preprocessed high-resolution image do
        Process the high-resolution image by Tesseract 4;
    end
    Rerank the 100 nearest high-resolution images by TF-IDF cosine similarity
     between the Google Vision AI ground truth and the Tesseract 4 output;
    if the nearest high-resolution image is the same after reranking then
        Prelink the low-resolution image with its nearest high-resolution image;
    end
end
foreach book do
    if all low-resolution images in the book are prelinked then
        Link all low-resolution images in the book with their prelinked
         high-resolution images;
    end
end
```

3.2 Quantitative Evaluation

Speed To evaluate the speed of the OCR algorithms, we measured and report the wall clock time in days to process the test dataset on a single CPU/GPU. For Tesseract, we used the `apollo`, `asteria04`, `mir`, `hypnos1`, `turnus01`, and `nymfe{23..74}` nodes at the Faculty of Informatics, Masaryk University, totalling 492 CPU cores. For OCR-D, we used the the `epimetheus{1..4}` nodes for evaluating the CPU speed, and `turnus03` for evaluating the GPU speed. For OCR-D, we also measured and report the speed of each workflow step separately.

Accuracy To evaluate the accuracy of the OCR algorithms, we measured and report the Character Error Rate (CER) and the Word Error Rate (WER): [8]

$$\text{ErrorRate}(A, B) = \frac{\text{EditDistance}(A, B)}{\text{Maximum}(|A|, |B|)} \times 100\%.$$

For WER, we lower-cased and deaccented the texts in addition to tokenization in order to better model our full-text search use case. Although CER and WER are correlated, we were mainly interested in WER, which is harder and better corresponds to our full-text search use case.

To discover the origin of the ground truth, we evaluated the accuracy of the Google Cloud Vision AI using both the low-resolution images (f.k.a. G-low) and the high-resolution images (f.k.a. G-high), not just the high-resolution images.

3.3 Qualitative Evaluation

To better understand the strengths and weaknesses of the individual OCR algorithms, we compared their outputs on three different scanned images:

1. A random image from the test dataset.
2. The image with the best WER using the least accurate OCR algorithm.
3. The image with the worst WER using the most accurate OCR algorithm.

To inspect the quality of the ground truth, we manually classified the differences between the ground truth and the OCR output on the three images as either improvements or errors, and we report the ratio of improvement to error.

4 Results

In this section, we will describe the results of the evaluation described in the previous section. We report both quantitative and qualitative evaluation results.

4.1 Quantitative Evaluation

Speed Table 1 shows that Tesseract 3 is the fastest OCR algorithm. The second fastest Tesseract 4 is more than twice as slow as Tesseract 3 because of the higher computational complexity of the non-GPU-accelerated LSTM-based OCR engine. OCR-D achieves speed that is comparable with Tesseract 4 despite the GPU acceleration of Calamari, which is likely because of the fully-configurable workflows and the high disk I/O caused by the locking and the common updates

Table 1. The wall clock time to process the test dataset with different OCR algorithms ordered from the fastest to the slowest. Since Google Cloud Vision AI is not self-hosted, it was not included in the speed evaluation.

	Tesseract 3	Tesseract 4	OCR-D (GPU)	Tesseract 3 + 4	OCR-D (CPU)
Time (days)	*61.12*	127.69	140.39	172.11	174.07

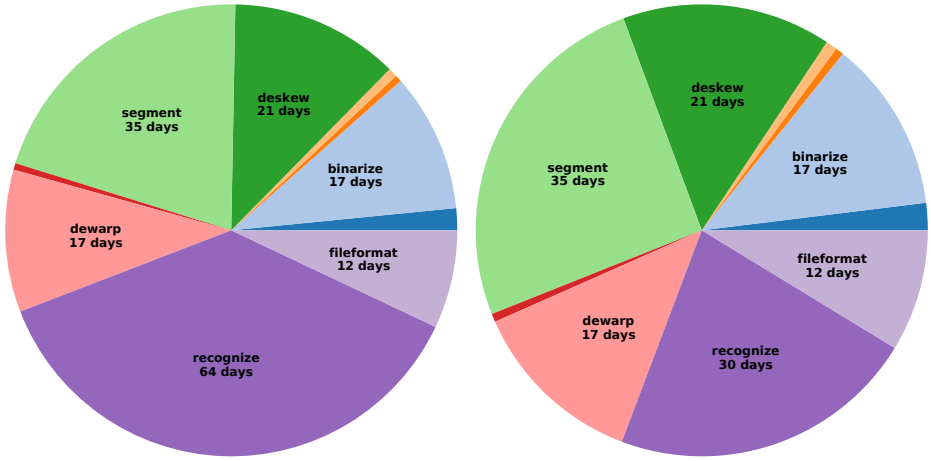


Fig. 1. The wall clock time to process the test dataset on the level of the individual workflow steps of OCR-D on a single CPU (left) and on a single GPU (right).

to a large METS XML document containing the history of all workflow steps executed for all scanned images in a book. Tesseract 3 + 4 is the second slowest with the time roughly equivalent to the sum of the times of Tesseract 3 and Tesseract 4. OCR-D without GPU acceleration is the slowest OCR algorithm.

Figure 1 shows that without GPU acceleration, more than one third of OCR-D's time is taken up by the Calamari line OCR workflow step. GPU acceleration causes more than 2× speed increase of this workflow step, whereas the other steps do not benefit from GPU acceleration.

Accuracy Table 2 shows that Google Cloud Vision AI achieved perfect accuracy with neither low-resolution nor high-resolution images, which leads us to the conclusion that the ground truth had been produced using the less appropriate `TEXT_DETECTION` feature. The higher accuracy of Google Cloud Vision AI with low-resolution images leads us to believe that the ground truth had been produced using the low-resolution images rather than the high-resolution images. Both findings raise doubts about the quality of the ground truth.

Table 2 also shows that Tesseract 4 is the most accurate OCR algorithm with the second most accurate being Tesseract 3. Ensembling Tesseract 3 and Tesser-

Table 2. The accuracy of different OCR algorithms on the test dataset ordered from the most accurate to the least accurate. G-low and G-high correspond to the Google Cloud Vision AI with low-resolution images and high-resolution images, respectively.

	G-low	G-high	*Tesseract 4*	Tesseract 3	Tesseract 3 + 4	OCR-D
WER (%)	3.85	5.24	*13.77*	16.04	18.70	30.94
CER (%)	2.44	3.38	*9.81*	10.60	12.00	19.88

act 4 leads to the second worst accuracy, which suggests a poor confidence heuristic of Tesseract in selecting words from the suggestions of both OCR algorithms. OCR-D is the least accurate OCR algorithm, likely because we used the recommended Calamari models trained on German Fraktur,[11] whereas most of our images are scans of letterpress prints in Antiqua typefaces with Czech accents.

4.2 Qualitative Evaluation

Random image We randomly selected page 5 from *Archiv český 01*,[12] with the following ground truth text:

> I. PSANJ ČESKÁ CJSARE SIGMUNDA
> od roku 1414 do 1437.
>
> Panu Čeńkowi z Wartenberka a z Weselj, neyw. purkrabj Pražskému: aby královně Sofii na statejch gegjch wěnnjch nátisku činiti nedopauštěl.
>
> Z Teplice u Ferrary, 1414, 24 Mart.
>
> Sigmund z božie milosti Rimský a Uherský oc. král.
>
> Urozený wěrný mily! Slyšíme, že někteří páni w Čechách najoswiecenější kněžnu pant Sofii, královnu Českú, sestru naši milú, mienie a chystaji sie, jie na jejím wěně mimo práwo a mimo panský nález tisknúti; jenžto nerádi slyšíme, anižbychom toho rádi dopustili, by sie jie to od koho mělo státi. Protož od tebe žádáme i prosime, byloliby żeby jmenowanú královnu, sestru naši milú, mimo prawo kto tisknúti, a nebo na jejiem wěně překážeti chtěl, aby podle ní stál, a jie wěrně pro nás pomohl, aby od swého nebyla tištěna. Na tom nám zwláštní službu učiniš a ukážeš. Dán w Teplici u Ferrarii, weder matky božie Annuntia-tionis, léta královstwie našich Uherského uc. w XXVII., a Římskéhow čtwrtém létě,
>
> Ad mandatum D. Regis: Michael de Priest.
>
> Urozenému Čeńkowi z Wesele
> wěrnému nám zwláště milému.

Tesseract 3 achieved 6.36% WER and 3.96% CER. Over 42% of changes were improvements, raising doubts about the quality of the ground truth:

>]. PSANJ ČESKÁ CJSAŘE SIGMUNDA
> od roku 1414 do 1437.
>
> Panu Čeňkowi z Wartenberka a z Weselj, neyw. purkrabj Pražskému: aby královně Sofii na. statcjch gegjch wénnjch nátisku činiti nedopauštěl.
>
> Z Teplice uFermry, 1414, 24 Mart.
>
> Sigmund z božie milosti Římský a Uherský oc. král.
>
> Urozený wěrný milý! Slyšíme, že někteří páni w Čechách najoswiecenější kněžnu paní Sofii, královnu Českú, sestru nasi milů, mienie a chystají sie, jie na jejím weně mimo práwo a mimo panský nález tisknúti; jenžto nerádi slyšíme, anižbychom toho rádi dopustili, by sie jie to od koho mělo státi. Protož od tebe žádáme i prosime, byloliby żeby jmenowanů královnu, sestru naši milů, mimo prawo kto tisknúti, a nebo na jejiem weně překážeti chtěl, aby podlé ní stál, a jie wěrně pro nás pomohl, aby od swého nebyla tištěna. Na tom nám zwláštní službu učiniš & ukážeš. Dán W Teplici u Ferrarii, weċer matky božie Annuntia-tionis, létá královstwie našich Uherského oc. w XXVII., & Římského w čtwrtém létě.
>
> Ad mandatum D. Regis: Michael de Priest.
>
> Urozenému Čeňkowi z Weselé
> Wěrnému nám zwláště milému.

Tesseract 4 achieved 8.57% WER, 4.95% CER. 28% changes were improvements:

> I. PSANJ ČESKÁ CJSAŘE SIGMUNDA
> od roku 1414 do 1437.
>
> Panu Čeňkowi z Wartenberka a z Weselj, neyw. purkrabj Pražskému: aby královně Sofii na statcjch gegjich wénnjch nátisku činiti nedopauštěl.
>
> Z Teplice u Ferrary, 1414, 24 Mart.
>
> Sigmund z bozie milosti Římský a Uherský oc. král.
>
> Urozený wěrný milý! Slyšiíme, že někteří páni w Čechách najoswiecenější kněžnu paní Sofii, královnu Českú, sestru naši milů, mienie a chystají sie, jie na jejím wčeně mimo práwo a mimo panský nález tisknůti; jenžto nerádi slyšíme, anižbychom toho rádi dopustili; by sie jie to od koho mělo státi. ProtoZ od tebe žádáme i prosime, byloliby Zeby jmenowanů královnu, sestru naài milů, mimo práwo kto tisknůti, a nebo na jejiem wéné prekáżeti chtěl, aby podlé ni stàl, a jie wěrně pro nás pomohl, aby od swého nebyla tištěna. Na tom nám zwláštní sluZbu uciniS a ukážeš. Dán w Teplici u Ferrarii, weċer matky boZie Annuntia-tionis, léta královstwie našich Uherského sc. w XXVII., a Římského w čtwrtém létě,
>
> Ad mandatum D. Regis: Michael de Priest.
>
> Urozenému Ceünkowi z Weselé
> wérnému nááàm zwlášté milému.

[11] https://ocr-d-repo.scc.kit.edu/models/calamari/GT4HistOCR/model.tar.xz
[12] https://sources.cms.flu.cas.cz/src/index.php?cat=10&bookid=792&page=5

Tess. 3 + 4 achieved 12.43% WER, 8.35% CER. 12% changes were improvements:

I. PSANJ ČESKÁ CJSAŘE SIGMUNDA
od roku 1414 do 1437.

Panu Čeňkowi z Wartenberka a z Weselj, neyw. purkrabj Pražskému: aby královně Sofii na statcjch gegjch wěnnjch nátisku činiti nedopauštěl.

Z Teplice u Ferrary, 1414, 24 Mart.

Sigmund z božie milosti Římský a Uherský oc. král.

Urozeny wörny mily! Sly&tme, ze nékterí páni w Cechách najoswiecenéjsi knéźnu pani Sofii, králewnu Ceskü, sestru nasi milü, mienie a chystaji sie, jie na jejim wéné mimo práwo a mimo pansky nález tisknüti; jenZto nerádi slysime, aniZbychom toho rádi dopustili, by sie jie to od koho mélo státi. ProtoZ od tebe Zádáme i prosime, byloliby Zeby jmenowanü králewnu, sestru naài milü, mimo práwo kto tisknüti, a nebo na jejiem wéné prekázeti chtél, aby podlé ni stàl, a jie wérné pro nás pomohl, aby od svého nebyla tisténa. Na tom näm zwlästni sluZbu ucini$ a ukàáZes. Dáàn w Teplici u Ferrari, wecer matky bozie Annuntia-tionis, léta králowstwie nasich Uherského oc. w XXVIL, a Rimskeho w ótwrtém lété,

Ad mandatum D. Regis: Michael de Priest.

Urozenému Ceünkowi z Weselé
wérnému náàm zwlásté milému.

OCR-D achieved 27.17% WER, 13.95% CER. Only 5% of changes were improvements, which indicates that the ground truth can still differentiate good and bad OCR outputs. Due to the training on German Fraktur without Czech accents, most errors originate from accented letters:

J. PaAN SINA
od roku 1414 do 1437.

Panu Cekowi 2 artenberka a 2 Veselj, neyw. purkrabj Praiskému: aby krâlown Sofii na statejch gegjeh wènnjeh nâtisku öiniti nedopaustël.

Teplice u Ferrar, 1414, 24 Mart.

Sigmund z boie milosti Rimsky a Uhersk oc. krâl.

ſrozeny wörny mily! Slysime, e nekteri pâni w Cechâch najoswiecenèjsi knênu pani Sofii, kralewnu Cesk, sestru nasi milu, mienie a chystaji sie, jie na jejim wenè mimo prâwo a mimo pansky nalez tisliti; jenâto nerâdi slysime, anizbychom toho radi dopustili, by sie jie to od koho mêlo stâti. Proto od tebe adame i prosime, byloliby eby jmenowanu kralewnu, sestru nasi milu, mimo prâwo kto tiskniti, a nebo na jejiem wènè prekâieti chtél, aby podlé ni stal, a jie wvrnè pro nas pomohl, aby od svého nebyla tistêna. Na tom nâm zwlastni sluibu ucinis a ukâes. Dân w Teplici u Ferrarii, wecer matky boie Annuntia-tionis, léta krâlowstwie nasich Uherského oe. w XXVJ., a Rimského w êtwrtém lètè.

Ad mandatum D. Regis: Michael de Priest.

Urozenému Ceükowi 2 Veselé
wêrnému nam zlastè milému.

Best image OCR-D achieved the best accuracy (0% WER, 0.29% CER) on page 59 of *CIM I*.[13] None of the changes were improvements, confirming our thesis that the ground truth can reliably distinguish good and bad OCR outputs:

vllam exigere, petere aut recipere volumus pecuniam, aut exigi, peti
…
vngelta in dicta ciuitate videlicet pannorum, mercium institarum et braxaturas ceruisie cum prouentibus ipsorum iuxta placita, per nos et ipsos ciues nostros hincinde habita, infra spacium dictorum annorum cum adicione quinti anni pro se recipere debeant et habere, quousque omnia et singula debita per ipsos ciues nostros quocumque modo contracta in integrum fuerint persoluta, impedimento nostro et cuius- libet non obstante. Debet quoque ipsum vngeltum sic recipi atque dari, videlicet quod vendens pannos siue merces institarum ciuis vel hospes de qualibet sexagena grossorum Sex paruos denarios vsuales et emens merces easdem totidem paruos in prima vendicione et em- pcione tantum et non vlterius soluere est astrictus, et quilibet braxans ceruisiam in ipsa ciuitate de vna braxatura ceruisie vnum grossum pro vngelto ipsis ciuibus dare debet. Si quis vero sub vna sexagena grossorum vendiderit vel emerit in mercibus, vt predicitur, quidquam, de huiusmodi vendicione et empcione pro vngelto nichil dabit. Addi- cimus eciam, quod omnes mercatores Pragam cum pannis quibus- cunque uel mercibus institarum ibidem non emptis transire volentes,
…
Liber vetustissimus statutorum c. 993 str. 61 V archivu m. Prahy.

Worst image Tesseract 4 achieved the worst accuracy (100% WER, 81.56% CER) on page 1297 of *CIM II*.[14] All changes were improvements, since the ground truth did not detect two-column page layout, unlike Tesseract 4. This confirms our thesis that although the ground truth can distinguish good and bad OCR outputs, it cannot distinguish OCR outputs that are better than the ground truth.

[13] https://sources.cms.flu.cas.cz/src/index.php?cat=12&bookid=117&page=230
[14] https://sources.cms.flu.cas.cz/src/index.php?cat=12&bookid=118&page=1327

5 Conclusion and Future Work

In our work, we have compared the speed and the accuracy of six OCR algorithms on the CMS online dataset from the AHISTO project.

Based on our results, we conclude that the ground truth OCR outputs in our dataset are low-quality and should be replaced or supplemented either by human judgements, or by synthetic data produced from searchable PDF documents.

As far as the ground truth can be trusted, Tesseract 4 is the second fastest and the most accurate OCR algorithm, which also detects language at the level of words. Pre-detecting the language of paragraphs using n-gram frequency analysis and then using Tesseract 4 only with the language models corresponding to the detected languages can further improve its accuracy. [3, Section 4.4]

OCR-D with the Calamari line OCR is comparable to Tesseract 4 in speed and is likely to produce more accurate results than Tesseract 4. However, the performance of Calamari was undermined by its poor pre-trained models. Additionally, Calamari does not detect language. However, OCR-D can align OCR outputs from Calamari and Tesseract 4, which can bring the best of both worlds.

Applying super-resolution algorithms [5] to the low-resolution scanned images can make them suitable for OCR and the AHISTO project.

Acknowledgments. First author's work was funded by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project and also by TAČR Éta, project number TL03000365. I also sincerely thank my colleagues from the NLP Centre and the MIR research group at the Masaryk University for their insight.

References

1. Breuel, T.M.: The hOCR microformat for OCR workflow and results. In: ICDAR 2007. vol. 2, pp. 1063–1067. IEEE (2007)
2. Neudecker, C., Baierer, K., Federbusch, M., Boenig, M., Würzner, K.M., Hartmann, V., Herrmann, E.: OCR-D: An end-to-end open source OCR framework for historical printed documents. In: Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage. pp. 53–58 (2019)
3. Panák, R.: Digitalizace matematických textů. Master's thesis, Faculty of Informatics, Masaryk University (2006), https://is.muni.cz/th/pspz5/
4. Reul, C., Springmann, U., Wick, C., Puppe, F.: State of the art optical character recognition of 19th century fraktur scripts using open source engines (2018), https://arxiv.org/pdf/1810.03436.pdf
5. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
6. Smith, R.: An overview of the Tesseract OCR engine. In: ICDAR 2007. vol. 2, pp. 629–633. IEEE (2007)

7. Smith, R., Antonova, D., Lee, D.S.: Adapting the Tesseract open source OCR engine for multilingual OCR. In: Proceedings of the International Workshop on Multilingual OCR. pp. 1–8 (2009)
8. 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)
9. Wick, C., Reul, C., Puppe, F.: Calamari-a high-performance tensorflow-based deep learning package for optical character recognition (2018), https://arxiv.org/pdf/1807.02004.pdf

Data Mining from Free-Text Health Records: State of the Art, New Polish Corpus

Krištof Anetta[1,2]

[1] Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
xanetta@fi.muni.cz
[2] Faculty of Arts, University of Ss. Cyril and Methodius
Nám. J. Herdu 2, Trnava, Slovakia
kristof.anetta@ucm.sk

Abstract. This paper deals with data mining from free-form text electronic health records both from global perspective and with specific application to Slavic languages. It introduces the reader to the promises and challenges of this enterprise and provides a short overview of the global state of the art and of the general absence of this kind of research in Central European Slavic languages. It describes pl_ehr_cardio, a new corpus of Polish health records with 18 years' worth of medical text. This paper marks the beginning of a pioneering research project in medical text data mining in Central European Slavic languages.

Keywords: EHR, electronic health records, named entity recognition, text data mining, NLP, natural language processing, Slavic languages, Polish.

1 Introduction

In recent years, as the performance of deep learning NLP approaches skyrocketed, a distinct niche of research has been gaining momentum: data mining from free-form text health records. In its short lifespan, it has already generated promising results when applied to English. This research extends the reach of this niche into Central European Slavic languages, where it has been largely absent. A large dataset of Polish health records spanning 18 years of data has been acquired and processed, forming the *pl_ehr_cardio* corpus. Apart from reviewing the general promises of data mining from health records and the global state of the art, this text also serves as an introduction to this cornerstone Polish corpus.

2 The Transformative Potential of Text-Mining Health Records

"A wealth of clinical histories remains locked behind clinical narratives in free-form text" [1] – this succinct sentence shows the key motivation behind text-mining health records. All over the world, there exist billions over billions of

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 13–22, 2020.

free-form text records of patient visits, hospitalizations, and other doctor-patient encounters, but most of this data is unstructured and allows very limited searching and processing (one estimate claims that 85 percent of actionable health information is stored in an unstructured way [2]) – all the valuable insights that big-data approaches can yield are inaccessible by default. When you consider the vastness of this database of human health right at our fingertips, the impulse to mine and structure the data is only natural (researchers have been urging development and collaboration in this niche [3]) – and with the rapid improvement of natural language processing using deep learning in the past years, mankind might finally possess the tools to do it. Civilization has always advanced on the shoulders of accumulated structured knowledge, and in the same vein, getting a grip on the world's health by leveraging decades of data with billions of cases might effect profound changes in medical science and global medical practice.

2.1 Statistics and Correlation

If properly processed into structured knowledge, databases of patient records would reveal crucial statistical information on diseases, including their early signs and effects of medicines, but also on various lifestyle-health correlations. Sample sizes, even as subsets after filtering for specific characteristics, would be orders of magnitude greater than in many contemporary clinical studies, and the researchers would be able to draw a dense network of interrelations between patient variables, symptoms, medications, and diagnoses. From the perspective of global health, standardized structured knowledge about various populations would make it easier for researchers to compare health and medical practice across the globe.

2.2 Evaluation and Prediction

An expert system leveraging health record databases as structured knowledge could also run calculations over the data and come up with entirely new judgments and estimations. It could:

- identify outliers and inconsistencies, which could discover either human error in diagnosis and prescription, or notable cases worthy of closer examination
- use deep learning to find patterns capable of predicting future outcomes of individual cases, which would open up avenues for risk group selection and better, more cost-efficient aiming of specific preventive measures.

3 Challenges

Free-form text health records exhibit several characteristics that make them more difficult to process than usual speech and writing. This is due to the hybrid nature of the text – it mixes codes and does not aspire to grammatical correctness or ease of comprehension, instead, efficiency is key and the requirement

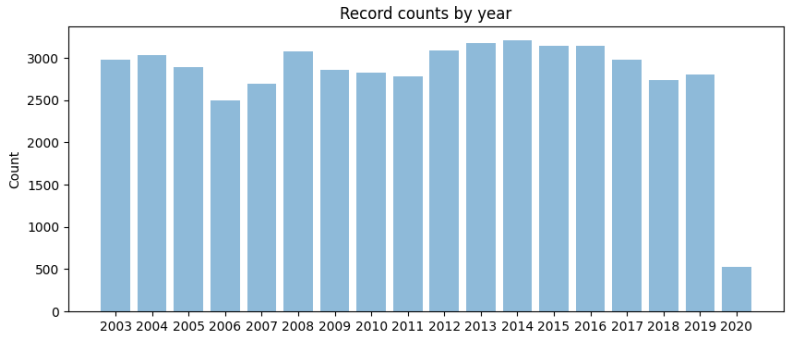


Fig. 1. Health records count distribution (*pl_ehr_cardio*)

for transparency is satisfied by being decipherable for a small group of medical experts in the respective language. The challenges involved in extracting structured data from health records include:

- Incompleteness: sentences in medical records may not correspond to standard sentence structure, missing essential syntactic elements or simply separating bits of information in a telegraphic fashion.
- Abbreviations: due to typing efficiency, abbreviating is very common, with many cases in which multiple abbreviated versions correspond to the same word.
- Bilingual text including Latin: since the meaning of medical text relies heavily on Latin words, the lexicon of the base language used for analysis needs to be extended with medical Latin. This issue has been described in [4].
- Numbers and codes: crucial information is encoded in measurements and symbolic representations, and a knowledge extraction system must be able to either determine their meaning based on form, unit or surrounding characters (such as "=" linking it to a variable), or at least recognize to which parts of surrounding language they are connected.
- Shifted or altered word meaning because of medical context: many words from natural language change their meaning in medical text or represent categorical variables, both nominal and ordinal.
- Typing errors: since electronic health records are often produced fast and interactively, many tokens are deteriorated, and the challenge in cases such as mistyped abbreviations is often too difficult even for human readers, requiring well-trained context-based solutions.

The above clearly demonstrates that for reliable and meaningful data extraction, existing NLP tools have to be heavily customized and very specifically trained.

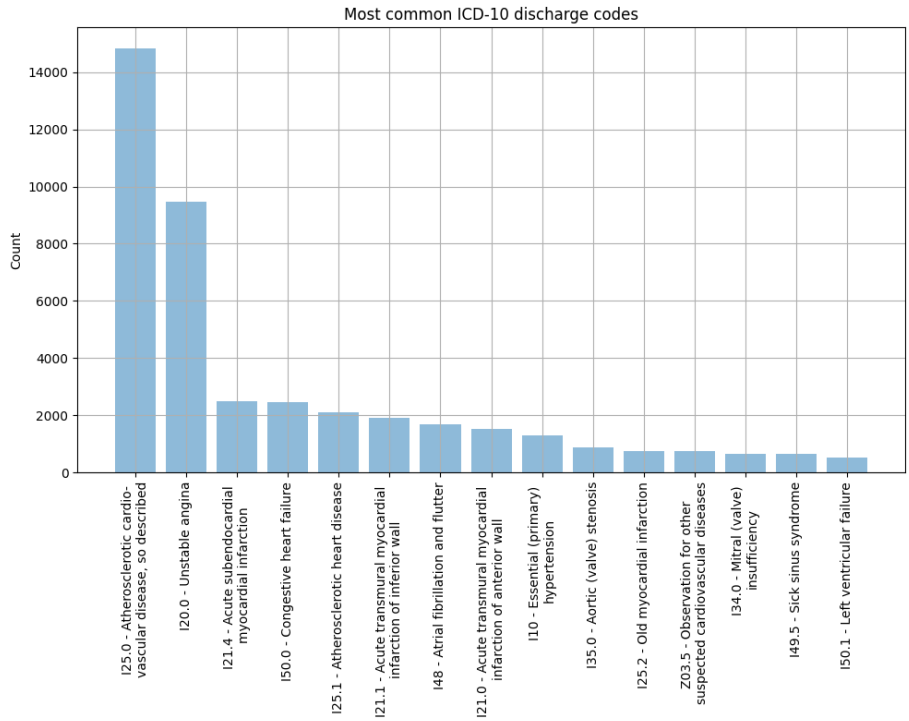


Fig. 2. Most common ICD-10 discharge codes (*pl_ehr_cardio*)

4 State of the Art, Prevailing Technology

4.1 Available Standard Frameworks

For English, several tools have been developed that directly or indirectly aid data extraction from free-form text health records:

- Apache cTAKES [5,6] is an open-source NLP system designed for extracting clinical information from the unstructured text of electronic health records. It is built using the UIMA (Unstructured Information Management Architecture) framework and Apache OpenNLP toolkit. Version 4.0 of this system was released in 2017. There have been attempts to adapt cTAKES for languages other than English [4], specifically Spanish [7] and German [8].
- MetaMap [6,9,10] is a program that maps biomedical text to the UMLS (Unified Medical Language System) Metathesaurus.

Standardized medical ontologies and term databases are an essential step towards comparability of results and interoperability – notable examples include:

- SNOMED CT [11,12] – multilingual clinical healthcare terminology containing codes, terms, synonyms, and definitions, considered to be the most com-

prehensive in the world. It has been employed in data extraction from health records [13].

4.2 Current Methods

Recent studies employing deep learning approaches have demonstrated that unstructured clinical notes improve prediction when added to structured data [14], similarly, the Deep Patient project has successfully included unstructured notes in its analyses [15]. Free-form text notes have proved especially useful for patient phenotyping [16]. Deep learning methods were also utilized in health record text mining for specific groups, such as those at risk of youth depression [17], prostate cancer [18], and the group of smokers [19], and also for adverse drug event detection [20,21,22].

Apart from various custom applications, convolutional neural networks [18,19], recurrent neural networks [20,22] and both uni- and bidirectional Long short-term memory (LSTM) [19,20,21,22] are notable candidates for the most widely used techniques. Some researchers also adapted BERT for clinical notes [23]. These deep learning architectures are frequently supplemented by the usage of conditional random fields (CRF) [21,22].

Overall, the tasks attempted with the above center around named entity recognition (NER) and relation extraction.

4.3 In Slavic Languages

Due to the relatively smaller size of Slavic languages, research related to them has been lagging global progress considerably, but there were notable attempts in Russian [24], Bulgarian [25], and Polish – since Polish is the subject of this research, it is worth noting the continuous efforts of a particular team [26,27,28], the most recent findings demonstrated on a corpus of more than 100,000 patient visits.

Table 1. Corpus statistics (*pl_ehr_cardio*)

Tokens	34,315,153
Words	23,831,785
Sentences	2,583,087
Average sentence length	9.226
Unique word forms	160,042
Unique word forms (lowercase)	141,685
Unique lemmas	124,727
Unique lemmas (lowercase)	114,556

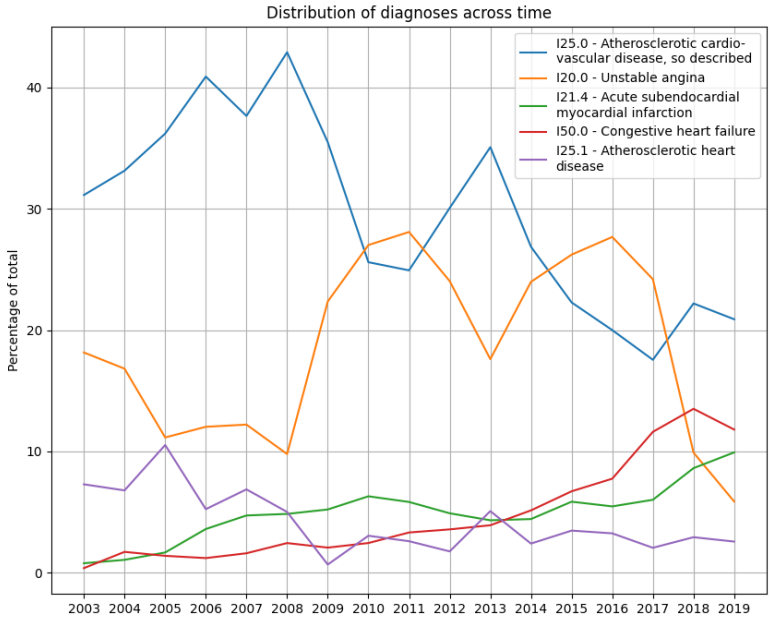


Fig. 3. Distribution of 5 most common discharge codes between 2003 and 2019 (*pl_ehr_cardio*)

5 New Corpus of Polish Health Records

The newly acquired dataset of Polish health records that forms the *pl_ehr_cardio* corpus consists of 50,465 recorded hospitalizations of cardiology patients, evenly distributed across the 17-year period between 2003 and 2019, also including partial data from 2020. Figure 1 demonstrates that years 2003 to 2019 are easily comparable in that no single year is overrepresented. After tagging these cardiology health records using the corpus management software tool SketchEngine [29][3], basic statistics were documented (Table 1)

Each record contains an ICD-10 discharge diagnosis code, which is a useful starting characteristic of the data. Figure 2 shows the most common discharge codes. Although the total number of records is quite evenly distributed over individual years, there is considerable variation in the proportions between discharge codes (Figure 3) – presumably in large part due to changes in diagnosing practice (e.g. preferred degree of specificity), although a more sensitive analysis

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

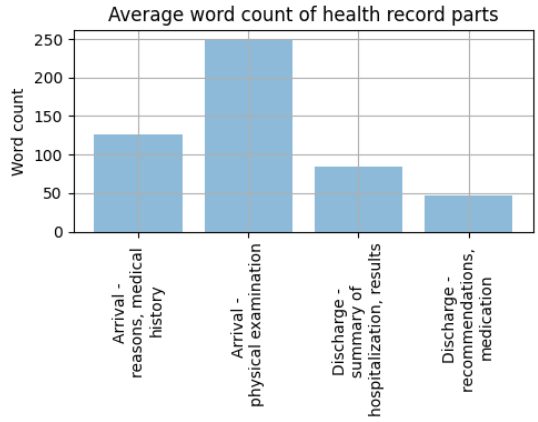


Fig. 4. Average word count of health record parts (*pl_ehr_cardio*)

might discover actual shifts in the occurrence of cardiological issues caused by shifting demographics and lifestyles.

Each record consists of 4 parts:

- Arrival: reasons, medical history (*Wywiad – Początek choroby*)
- Arrival: physical examination (*Wywiad – Badanie przedmiotowe*)
- Discharge: summary of hospitalization, results (*Epikryza – Badanie fizykalne*)
- Discharge: recommendations, medication (*Epikryza – Zalecenia lekarskie*)

Every part is written in a different style and concentrating on different concepts, and will require custom-tailored attention. Figure 4 shows that roughly half of the available text is concentrated in part 2, which is concerned with the physical examination after the patient's arrival. Word count may not exactly correspond to the amount of information present, but it gives a rough indication of the profile of the data, among others that there is ample information about symptoms and physical examination findings, which is especially valuable when correlated with diagnoses. Also, part 4 containing recommendations and medication prescriptions is usually written in a much more condensed fashion, which means that its relatively lower average word count still provides generous amounts of data on medication.

From the various challenges mentioned in the first sections, this Polish corpus does not suffer from too much Latin usage or abbreviation, but the syntax of its sentences very often leaves out elements (notably verbs) and punctuation, which complicates dependency parsing. After preliminary processing using the spaCy [30] library with pl_core_news_lg model (500,000 unique vectors), it has become obvious that named entity recognition trained on regular Polish corpora yields no useful results and it will require specific training. On the other hand, spaCy's dependency parsing correctly identified a large portion of dependencies such as nominal subject, nominal modifier, or adjectival modifier,

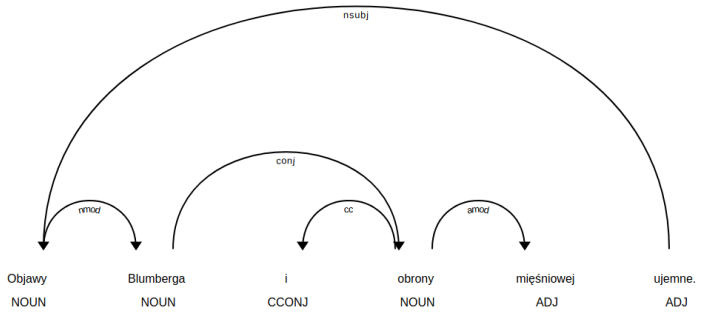


Fig. 5. Example sentence with dependencies shown in spaCy's displaCy visualizer

which will be crucial in extracting information about physical examination findings.

6 Conclusion

This paper's purpose has been twofold. First, it aimed to briefly introduce the growing niche of data mining from the unstructured text of health records including the promises, challenges, and current state of the art in this area. Arguably, this niche's growth is still in the beginnings, given the magnitude of existing data and the centrality of big data approaches to a case study-based science like medicine. For decades, this enterprise has been viewed as a major opportunity for the expansion of medical knowledge and practice, but only the advent of highly effective deep learning NLP methods did bring sufficient power to fully leverage the heaps of unstructured content.

Second, this paper used the opportunity to describe a newly formed corpus of Polish health records and thereby demonstrate some of the ideas and considerations in beginning such research on a concrete example. This dataset detailing more than 50,000 cardiology hospitalizations over 18 years will be the subject of subsequent studies, in which it will both pioneer a topic rarely broached in Slavic languages and contribute valuable descriptive and correlational information about cardiology patients, their symptoms, procedures, medication, and diagnoses to the medical community.

Acknowledgments. 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.

This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

1. Sheikhalishahi, S., Miotto, R., Dudley, J. T., Lavelli, A., Rinaldi, F., Osmani, V.: Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review. *JMIR Medical Informatics* 7(2):e12239, (2019). https://doi.org/10.2196/12239
2. Nenadic, G.: Key Patient Information Stored in Routinely Collected Healthcare Free-text Data is Still Untapped. *Open Access Government* (2019).
3. Ohno-Machado, L.: Realizing the Full Potential of Electronic Health Records: The Role of Natural Language Processing. *Journal of the American Medical Informatics Association* 18(5), 539 (2011). https://doi.org/10.1136/amiajnl-2011-000501
4. Névéol, A., Dalianis, H., Velupillai, S. et al.: Clinical Natural Language Processing in Languages Other than English: Opportunities and Challenges. *Journal of Biomedical Semantics* 9(12), (2018). https://doi.org/10.1186/s13326-018-0179-8
5. Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., Chute, C. G.: Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, Component Evaluation and Applications. *Journal of the American Medical Informatics Association* 17(5), 507–513 (2010). https://doi.org/10.1136/jamia.2009.001560
6. Reátegui, R., Ratté, S.: Comparison of MetaMap and cTAKES for Entity Extraction in Clinical Notes. *BMC Medical Informatics and Decision Making* 18(74), (2018). https://doi.org/10.1186/s12911-018-0654-2
7. Costumero, R., García-Pedrero, A., Gonzalo-Martín, C., Menasalvas, E., Millan, S.: Text Analysis and Information Extraction from Spanish Written Documents. In: Slezak, D., Tan, A. H., Peters, J., Schwabe, L. (eds.) *Brain Informatics and Health. BIH 2014. Lecture Notes in Computer Science*, vol 8609, pp. 188–197. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09891-3_18
8. Becker, M., Böckmann, B.: Extraction of UMLS® Concepts Using Apache cTAKES™ for German Language. *Studies in Health Technology and Informatics* 223, 71-76 (2016).
9. Aronson, A. R.: Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. *Proceedings, AMIA Symposium*, pp. 17–21 (2001).
10. Aronson, A. R., Lang, F.: An Overview of MetaMap: Historical Perspective and Recent Advances. *Journal of the American Medical Informatics Association* 17(3), 229–236 (2010). https://doi.org/10.1136/jamia.2009.002733
11. Donnelly, K.: SNOMED-CT: The Advanced Terminology and Coding System for eHealth. *Studies in Health Technology and Informatics* 121, 279–290 (2006).
12. Benson, T., Grieve, G.: *Principles of Health Interoperability: SNOMED CT, HL7 and FHIR*. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30370-3
13. Peterson, K. J., Liu, H.: Automating the Transformation of Free-Text Clinical Problems into SNOMED CT Expressions. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 497–506 (2020).
14. Zhang, D., Yin, C., Zeng, J. et al. Combining Structured and Unstructured Data for Predictive Models: A Deep Learning Approach. *BMC Medical Informatics and Decision Making* 20, 280 (2020). https://doi.org/10.1186/s12911-020-01297-6
15. Miotto, R., Li, L., Kidd, B. et al.: Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports* 6, 26094 (2016). https://doi.org/10.1038/srep26094
16. Yang, Z., Dehmer, M., Yli-Harja, O. et al.: Combining Deep Learning with Token Selection for Patient Phenotyping from Electronic Health Records. *Scientific Reports* 10, 1432 (2020). https://doi.org/10.1038/s41598-020-58178-1

17. Geraci, J., Wilansky, P., de Luca, V. et al.: Applying Deep Neural Networks to Unstructured Text Notes in Electronic Medical Records for Phenotyping Youth Depression. *Evidence-Based Mental Health* 20, 83-87 (2017).
18. Leyh-Bannurah, S., Tian, Z., Karakiewicz, P. I., Wolffgang, U., Sauter, G., Fisch, M., Pehrke, D., Huland, H., Graefen, M., Budäus, L.: Deep Learning for Natural Language Processing in Urology: State-of-the-Art Automated Extraction of Detailed Pathologic Prostate Cancer Data From Narratively Written Electronic Health Records. *JCO Clinical Cancer Informatics* 2, 1-9 (2018) https://doi.org/10.1200/CCI.18.00080
19. Rajendran, S., Topaloglu, U.: Extracting Smoking Status from Electronic Health Records Using NLP and Deep Learning. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, pp. 507–516 (2020).
20. Wunnava, S., Qin, X., Kakar, T., Sen, C., Rundensteiner, E. A., Kong, X.: Adverse Drug Event Detection from Electronic Health Records Using Hierarchical Recurrent Neural Networks with Dual-Level Embedding. *Drug Safety* 42(1), 113-122 (2019). https://doi.org/10.1007/s40264-018-0765-9
21. Christopoulou, F., Tran, T. T., Sahu, S. K., Miwa, M., Ananiadou, S.: Adverse Drug Events and Medication Relation Extraction in Electronic Health Records with Ensemble Deep Learning Methods. *Journal of the American Medical Informatics Association* 27(1), 39-46 (2020). https://doi.org/10.1093/jamia/ocz101
22. Yang, X., Bian, J., Gong, Y. et al.: MADEx: A System for Detecting Medications, Adverse Drug Events, and Their Relations from Clinical Notes. *Drug Safety* 42, 123–133 (2019). https://doi.org/10.1007/s40264-018-0761-0
23. Huang, K., Altosaar, J., Ranganath, R.: ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. (2019) arXiv:1904.05342
24. Gavrilov, D., Gusev, A., Korsakov, I.: Feature Extraction Method from Electronic Health Records in Russia. In: *Proceeding of the 26th Conference of FRUCT Association*, pp. 497-500 (2020).
25. Zhao, B.: Clinical Data Extraction and Normalization of Cyrillic Electronic Health Records Via Deep-Learning Natural Language Processing. *JCO Clinical Cancer Informatics* 3, 1-9 (2019). https://doi.org/10.1200/CCI.19.00057
26. Mykowiecka, A., Marciniak, M., Kupsc, A.: Rule-based Information Extraction from Patients' Clinical Data. *Journal of Biomedical Informatics* 42(5), 923-936 (2009). https://doi.org/10.1016/j.jbi.2009.07.007
27. Marciniak, M., Mykowiecka, A.: Terminology Extraction from Medical Texts in Polish. *Journal of Biomedical Semantics* 5(24), (2014). https://doi.org/10.1186/2041-1480-5-24
28. Dobrakowski, A. G., Mykowiecka, A., Marciniak, M., Jaworski, W., Biecek, P.: Interpretable Segmentation of Medical Free-Text Records Based on Word Embeddings. In: Helic, D., Leitner, G., Stettinger, M., Felfernig, A., Raś, Z. W. (eds.) *Foundations of Intelligent Systems. ISMIS 2020. Lecture Notes in Computer Science*, vol 12117. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59491-6_5
29. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: Ten Years On. *Lexicography* 1, 7-36 (2014).
30. Honnibal, M., Montani, I.: *spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing*. (2017)

Efficient Management and Optimization of Very Large Machine Learning Dataset for Question Answering

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. Question answering strategies lean almost exclusively on deep neural network computations nowadays. Managing a large set of input data (questions, answers, full documents, metadata) in several forms suitable as the first layer of a selected network architecture can be a non-trivial task. In this paper, we present the details and evaluation of preparing a rich dataset of more than 13 thousand question-answer pairs with more than 6,500 full documents. We show, how a Python-optimized database in a network environment was utilized to offer fast responses based on the 26 GiB database of input data. A global hyperparameter optimization process with controlled running of thousands of evaluation experiments to reach a near-optimum setup of the learning process is also explicated.

Keywords: question answering; dataset management; machine learning; optimization

1 Introduction

Current hardware and software architectures for neural network computations are capable of processing tens of thousands of input data units relatively fast, especially in a situation of distributed processing. However, a bottleneck of such processing can lie in copying the input data between the computing machines. Imagine a set of hundreds of possible answers to a question with each answer as a set of 500-dimensional word vectors including a selected broader context of the answer. Then in each training epoch the computations need to engage thousands of such questions from the training set. In case of inefficient storage and data transfer, such dataset can occupy hundreds of gigabytes which can, of course, negatively influence the training process running time.

In this paper, the details of the data processing in the answer selection task with the Simple Question Answering Database (SQAD [1,2]) are presented. The training and testing process is further enhanced by semi-automatized search of optimal hyperparameters setup.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 23–34, 2020.

1.1 Data Processing in Related Works

Question answering (QA) systems are particularly developed for the mainstream languages where a number of datasets are published. A well-known example of such dataset is the Stanford Question Answering Database (SQuAD [3]) that was used in more than 80 state-of-the-art works.

In [4], Patel et al enriched the Stanford's JSON formatted data with sentence embeddings (trained on large English corpus) using the InferSent tool [5] by the Facebook research team. The semantic representations of the sentences are then used to evaluate semantic distance between possible answers and questions.

In [6], Park presented indexing all words inside the Stanford database into an internal vocabulary and represented all SQuAD records as a list of word indexes to the internal vocabulary. Finally, Park enriched all words within the vocabulary by GloVe [7] word vectors.

In [8], Tiutiunnyk decided to store a new Ukrainian QA dataset in the PostgreSQL database to allow their system direct access to all the dataset records.

1.2 ZODB Database System

ZODB[1] is a transparent Python-object persistence (database) system that is able to store selected data models (especially classes and objects in the Python source codes). With comparison to standard relational database systems such, the ZODB database system is able to store huge hierarchical structures that are not limited to one data type. Hierarchical databases on the other hand do not support transactions and are bound with all the restrictions resulting from the relational model.

In the current implementation of the Czech SQAD database, the ZODB system is used as the main storage of all the dataset data. This approach differs from the standard approaches used with the Stanford SQuAD database mentioned above. Details of the ZODB engagement are further presented in Section 2.

1.3 Hyperparameter Optimization

The most straightforward approach to neural network hyperparameter optimization is the standard *grid search* – a technique that involves manual choice of possible values for each optimized hyperparameter and generating all possible combinations. This can be computationally expensive as each added hyperparameter increases the number of required evaluations exponentially. Grid search is still available as an option in some major machine learning libraries (e.g. scikit-learn [9] or Optuna [10]).

Random search is designed to replace the exhaustive enumeration by taking random samples of hyperparameter values in the search space. The number of *trials* is typically limited by a pre-defined constant, see e.g. HyperOpt [11] or Optuna [10].

[1] `http://www.zodb.org/en/latest/`

Original text:
Ngoni (někdy též n'goni) je strunný hudební nástroj oblíbený v západní Africe. Někdy bývá označován jako primitivní předchůdce banja. Velikostí se ale podobá spíše ukulele. ... *[Ngoni (also called n'goni) is a string musical instrument popular in west Africa. It may be considered as a primitive predocessor of banjo. But according to its size it is more similar to ukulele. ...]*
Question:
Jakého typu je hudební nástroj ngoni? *[What kind of musical instrument is ngoni?]*
Answer:
strunný hudební nástroj *[string musical instrument]*
URL:
`https://cs.wikipedia.org/wiki/Ngoni`
Author:
login
Question type:
ADJ_PHRASE
Question type:
OTHER
Answer selection:
Ngoni (někdy též n'goni) je strunný hudební nástroj oblíbený v západní Africe. *[Ngoni (also called n'goni) is a string musical instrument popular in west Africa.]*
Answer extraction:
strunný hudební nástroj *[string musical instrument]*

Fig. 1. Example of the `SQAD` record No. 012878

The main downside of the aforementioned techniques is that they do not utilize the information from past trials. The *Sequential model-based optimization* (SMBO) tries to overcome this limitation by iteratively selecting the hyperparameters from the search space using a probabilistic model to minimise/maximise an objective function. The most commonly used probabilistic models are Tree-structured Parzen Estimators (in Optuna [10], HyperOpt [11], Ray-Tune [12]) and Gaussian processes (in GPyOpt [13]). One of the less commonly used models are Gradient Boosting (in scikit-learn [9]) and Random Forests (in TuneRanger [14]).

2 Managing Very Large Machine Learning Dataset

In this section, the implementation of the Czech SQAD dataset storage using the ZODB system is presented. ZODB allows fast access to the SQAD data and efficiently stores all data from the SQAD database raw records in the final form required by the AQA question answering system [15].

The ZODB database system is able to store Python objects without a need of extra format conversion, ZODB loads Python objects directly from the database.

word	lemma	tag
Ngoni	Ngoni	k1gNnSc1
(	(	kIx(
někdy	někdy	k6eAd1
též	též	k9
...		

Fig. 2. An example of the vertical format of POS-tagged text in SQAD.

word	NE tag
...	
přestoupil	O
do	O
Sparty	B
Praha	I
...	

Fig. 3. Example of *Link named entity* training data: `O` – regular word, `B` – beginning of named entity, `I` – continuation of named entity.

Currently, ZODB is used to store the complete SQAD database records.[2] In addition to raw POS-tagged texts, the SQAD database contains all important information derived from texts. In the source form [16], the SQAD database consists of several files per record. The data files within the SQAD database contain morphologically and lexically analyzed text in vertical format [3] that are converted into the SQAD-ZODB (fusion of SQAD data and ZODB database system) database.

Along with the original information stored in the SQAD database, the new SQAD-ZODB database contains several additional information that are important for different modules inside the AQA system. These additional features are automatically computed from the original source data:

- *word vectors* – to boost the training procedure in the AQA answer selection module the new SQAD-ZODB database stores pre-computed word vectors pre-trained from large Czech corpora using the word2vec algorithm. In the current version, the SQAD-ZODB database stores 100-, 300- and 500-dimensional word vectors of each word. This allows a fast access to word vectors and flexibility in the training process of the answer selection module.
- *list of sentences containing exact answer* – during building SQAD-ZODB, the list of sentences that contain the exact answer is computed. This information is then used in the evaluation process of the answer selection module.
- *list of similar answers* – TF-IDF similarity scores between all sentences withing the record full text are computed. These scores are then used to fine tune

[2] See a SQAD record example in Figure 1.
[3] See Table 2 for an example.

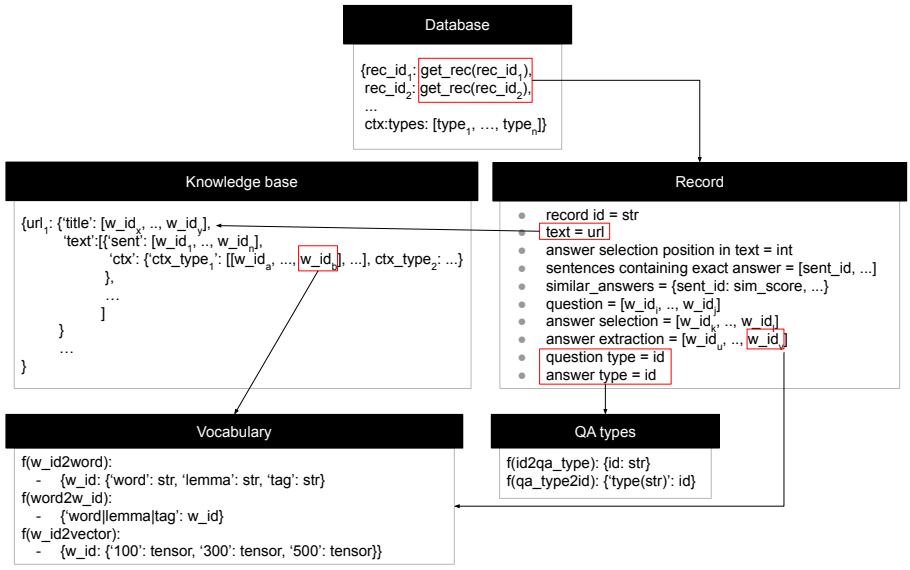


Fig. 4. The database structure of SQAD-ZODB.

the training setup of the answer selection module where the most similar negative sentences are used to boost the module ability to identify the correct answer within a list of very similar sentences.

– *answer context* – the task of identifying the main answer sentence is difficult mainly due to anaphoric expressions which "hide" the relevant entities by pronominal references. To supplement the neural network decision process, an information about the sentence context is provided to the answer selection module. To speed up the whole training process, the SQAD-ZODB database contains several types of context pre-computed from the original data. In the current version, the SQAD-ZODB database contains three types of contexts (more context types are to be added in the future work):
  - previous sentences – the context of N full sentences is added to each input article sentence.
  - phrases from previous sentences – using the rule-based SET parser [17], the system is able to identify all possible noun phrases within each sentence. M noun phrases from each of N preceding sentences are stored as the second context type.
  - "link named entities" from previous sentences form the third type of sentence context. See details in Section 2.1.

2.1 Link named entities

Link named entities (LNEs) are a specific type of standard named entities with regard to the information often expressed in questions and answers. LNEs are

defined as entities that are labeled with Wikipedia internal links. Inside each Wikipedia article, links that refer to other Wikipedia articles identify entities which are often significant in denoting an important piece of information. LNEs are identified in general texts by training a named-entity recognition (NER) system[4] with the whole Czech Wikipedia, where for each sentence all link named entities are marked, see Table 3 for an example. The final NER module is applied to the SQAD database and provides information about recognized link named entities which are used as a sentence context.

2.2 The SQAD-ZODB Architecture

The architecture of the current SQAD data in the ZODB database system is displayed in Figure 4. At the first level, the SQAD-ZODB database stores all records IDs and a function that builds the record content form 4 database parts (tables).

The *Record* object stores ten most critical information. Each *Record ID* is a unique identifier of a SQAD record.

The *Text* variable contains a unique *URL* that points to specific article inside the *Knowledge base* table. Thus for multiple records concerning one Wikipedia article, the database do not need to store that same article twice.

The *Answer selection position* stores an index of the sentence that contains the expected answer (used in the training part of the answer selection module).

Sentences containing exact answer is a list of sentence IDs that contain the exact answer (used in the training phase of the answer selection module where the system excludes these sentences as negative examples).

Similar answers is a list of similar sentences with their similarity scores (used to train the module to distinguish the correct answer within a list of very similar sentences).

Question, *answer selection*, *answer extraction* are lists of words IDs. Each word ID can be transformed into word, lemma, POS tag, 100-, 300- or 500-dimensional vector using the *Vocabulary table*.

The last two record features *question type* and *answer type* are also IDs pointing to specific question and answer type using the *QA types table*.

The *Knowledge base* table stores all articles used within the SQAD database. Avoiding duplicates and storing only list of the words IDs makes the knowledge base compact while maintaining all important information.

2.3 Updating the SQAD-ZODB Database

Due to the database transaction support in ZODB, updating the database is a straightforward task. After establishing a connection to the database, a user can add new records or add new features to existing records. In the SQAD development process, each new feature is a standalone script that can supplement the database with a single new feature of each record. The current transformation system consist of:

[4] The BERT-NER from `https://github.com/kamalkraj/BERT-NER` is currently used.

Table 1. Running times (in seconds) for a random sample of 100 queries. w – word, l – lemma, t – morphological tag, v1 – 100-dimensional vector, v3 – 300-dimensional vector, v5 – 500-dimensional vector.

Preloaded vocabulary						Not preloaded vocabulary					
init	12.44	w;l;t	1.63	w;l;t;v5	3.13	init	5.74	w;l;t	2.86	w;l;t;v5	5.05
w	13.21	w;l;v1	3.00	w;l;v1;v3	3.40	w	7.17	w;l;v1	4.58	w;l;v1;v3	5.43
l	2.02	w;l;v3	2.99	w;l;v1;v5	3.40	l	2.25	w;l;v3	4.58	w;l;v1;v5	5.45
t	1.42	w;l;v5	3.00	w;l;v3;v5	3.41	t	2.25	w;l;v5	4.59	w;l;v3;v5	5.47
v1	4.78	w;t;v1	3.00	w;t;v1;v3	3.40	v1	7.32	w;t;v1	4.75	w;t;v1;v3	5.46
v3	2.61	w;t;v3	3.03	w;t;v1;v5	3.42	v3	3.34	w;t;v3	4.63	w;t;v1;v5	5.47
v5	2.61	w;t;v5	2.58	w;t;v3;v5	3.01	v5	3.33	w;t;v5	4.59	w;t;v3;v5	5.45
w;l	1.53	w;v1;v3	3.30	w;v1;v3;v5	3.68	w;l	2.76	w;v1;v3	4.95	w;v1;v3;v5	5.77
w;t	1.95	w;v1;v5	3.28	l;t;v1;v3	3.40	w;t	2.41	w;v1;v5	4.95	l;t;v1;v3	5.47
w;v1	2.91	w;v3;v5	3.29	l;t;v1;v5	3.44	w;v1	4.11	w;v3;v5	4.95	l;t;v1;v5	5.49
w;v3	2.86	l;t;v1	3.01	l;t;v3;v5	3.41	w;v3	4.15	l;t;v1	4.60	l;t;v3;v5	5.45
w;v5	2.48	l;t;v3	3.01	l;v1;v3;v5	3.70	w;v5	4.10	l;t;v3	4.65	l;v1;v3;v5	5.76
l;t	1.94	l;t;v5	2.98	t;v1;v3;v5	3.71	l;t	2.75	l;t;v5	4.61	t;v1;v3;v5	5.76
l;v1	2.88	l;v1;v3	2.89	w;l;t;v1;v3	3.50	l;v1	4.11	l;v1;v3	4.94	w;l;t;v1;v3	5.87
l;v3	2.87	l;v1;v5	3.30	w;l;t;v1;v5	3.09	l;v3	4.11	l;v1;v5	4.96	w;l;t;v1;v5	5.92
l;v5	2.92	l;v3;v5	3.28	w;l;t;v3;v5	3.54	l;v5	4.13	l;v3;v5	4.98	w;l;t;v3;v5	5.89
t;v1	2.60	t;v1;v3	3.30	w;l;v1;v3;v5	3.80	t;v1	4.11	t;v1;v3	4.95	w;l;v1;v3;v5	5.88
t;v3	2.96	t;v1;v5	3.33	w;t;v1;v3;v5	3.81	t;v3	4.11	t;v1;v5	4.98	w;t;v1;v3;v5	6.25
t;v5	2.88	t;v3;v5	3.30	l;t;v1;v3;v5	3.81	t;v5	4.11	t;v3;v5	4.60	l;t;v1;v3;v5	6.24
v1;v3	3.08	v1;v3;v5	3.55	w;l;t;v1;v3;v5	4.03	v1;v3	4.24	v1;v3;v5	5.09	w;l;t;v1;v3;v5	6.86
v1;v5	3.06	w;l;t;v1	2.72			v1;v5	4.27	w;l;t;v1	5.08		
v3;v5	3.06	w;l;t;v3	3.08			v3;v5	4.22	w;l;t;v3	5.05		
total time					3m 38s	total time					5m 9s

- *sqad2zodb* transforms all records from the original SQAD source to the SQAD-ZODB database and adds pre-computed vectors to each word.
- *add_similar_sentences* enhances each record with the list of similar sentences.
- *add_sentences_containing_exact_answer* adds a list of sentences that contain the expected answer to the record.
- *context_previous_sentences* for each sentence in the article adds N preceding sentences as a context (where N is a user-defined parameter).
- *context_noun_phrases* for each sentence in the article adds M phrases for each of N preceding sentences.
- *context_ner* for each sentence in the article adds all link named entities recognized in N preceding sentences.

That is how each new record feature can be developed and tested separately.

2.4 SQAD-ZODB Performance

In the transformation process from SQAD to SQAD-ZODB, the database interface allows the training workers to transfer only those pieces of information that are required for the training process. The choice of the right information can greatly influence the data transfer running times. Table 1 summarizes the times needed to transfer different parts of the record.

The running times are proportional to the amount of data that need to be transferred. The efficiency of the ZODB storage is depicted by comparing the space requirements of the SQAD database in three formats are presented in Table 2.

2.5 SQAD-ZODB over Network

The SQAD-ZODB database is particularly used in the answer selection module. The training and hyperparameter optimization of this module requires a large amount of training setups to be tested. To speed up the training by distributing the process to multiple GPU-based servers, the SQAD-ZODB database was implemented within the ZEO[5] (Zope Enterprise Objects) library that allows to run the database in the client-server mode over network.

3 Large-scale Optimization of Machine Learning Hyperparameters

When training machine learning models, hyperparameter optimization is one of the key steps required to achieve acceptable performance. However, this process requires considerable efforts, especially in large search spaces of hyperparameter values. A variety of tools and libraries were developed to automate this process, employing sophisticated algorithms to achieve the task.

3.1 About Optuna

Optuna [10] is a relatively new hyperparameter optimization framework that aims to provide a simple setup for defining hyperparameter search spaces

[5] `http://www.zodb.org/en/latest/articles/old-guide/zeo.html`

Table 2. Disk usage for various storage methods. *Plain text* refers to the original plain text form of SQADv3 with all the pre-computed vectors. *Pickle* is a serialized dataset (using the Python `pickle` library) with only the necessary data to train models using the 500-dimensional embeddings.

Representation	Disk usage
Plain text	1,312.89 GB
Pickle	240.20 GB
SQAD-ZODB	25.08 GB

Framework	API Style	Pruning	Lightweight	Distributed	Dashboard	OSS
SMAC [3]	define-and-run	✗	✓	✗	✗	✓
GPyOpt	define-and-run	✗	✓	✗	✗	✓
Spearmint [2]	define-and-run	✗	✓	✓	✗	✓
Hyperopt [1]	define-and-run	✗	✓	✓	✗	✓
Autotune [4]	define-and-run	✓	✗	✓	✓	✗
Vizier [5]	define-and-run	✓	✗	✓	✓	✗
Katib	define-and-run	✓	✗	✓	✓	✓
Tune [7]	define-and-run	✓	✗	✓	✓	✓
Optuna (this work)	define-by-run	✓	✓	✓	✓	✓

Fig. 5. Comparison of hyperparameter optimization frameworks in terms of available features [10].

while being highly customizable. The parameter search spaces are defined using the *define-by-run* approach which allows the search spaces to be created and adjusted dynamically at runtime.

The Optuna framework is highly scalable from simple experimental computations to large-scale distributed optimizations. In order to accomplish this flexibility, Optuna supports many result storage forms like in-memory database, SQLite databases, or PostresSQL databases.

Optuna also includes a support for pruning algorithms that monitor intermediate values for the objective, and terminate a trial if a user-defined condition is not met. This premature termination is useful for saving time from unpromising trials.

3.2 Optuna and Answer Selection

In this section, we present improved results for the AQA answer selection task achieved using the Optuna library. Overall, 1,507 setups were trained in a fully automated fashion. In order to define the search space, a set of hyperparameters and their values was identified as affecting the model performance as displayed in Table 3.

The objective to maximize was the Mean Average Precision (MAP) of each trial. Out of the 1,506 trials, 455 were succesful (reached the evaluation on

Table 3. Hyperparameter values search space for the answer selection model

Hyperparameter name	Optuna distribution used	Range of values
BiGRU hidden size	discrete uniform	100-600 with the step of 20
Dropout	discrete uniform	0.0-0.6 with the step of 0.1
Batch size	categorical	1, 2, 4, 8, 16, 32, 64, 128, 256
Optimizer	categorical	Adam, Adagrad, Adadelta, SGD
Learning rate	logarithmic uniform	from 10^{-4} to 10^{-1}
Embedding dimension	categorical	300, 500

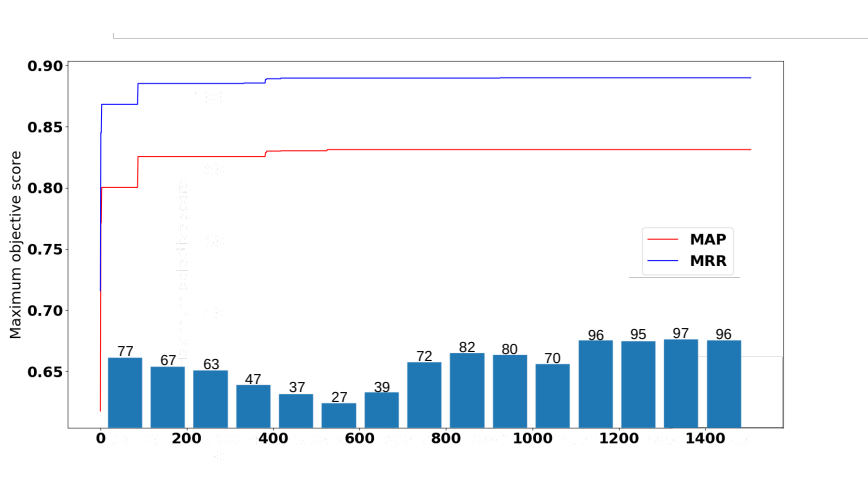


Fig. 6. Increase in MAP and MRR measured over 1,506 recorded runs. The histogram displays the amount of errors/prunes in the groups of 100.

test set), 365 were pruned, and 686 were errors due to GPU out-of-memory exception.

The most successful setup reached MAP of **83.13** and MRR of **88.99**, which is an increase of **0.8** percent when compared to last published result (MAP of **82.33**). The best trial's setup uses the embedding dimension of 300, BiGRU hidden size of 520, the dropout probability of 0.3, the batch size of 4 (with each sentence in the document used), the Adagrad optimizer with the learning rate set to 0.0042.

If we compare the result with the best model setup that uses the 300-dimensional embeddings, the current best setup has achieved an increase of **1.15**% (versus MAP of **81.92**). Unfortunately due to many erroneous trials with 500-dimensional word embeddings, Optuna's search was more focused in optimization with 300-dimensional embeddings instead.

4 Conclusions and Future Work

In the paper, we have presented the details of managing the efficient storage and data transfer of very large question answering dataset. The implementation using the ZODB database framework allows fast data distribution for network-based training and hyperparameter optimization computations.

Using the Optuna parameter optimization framework we have achieved a 0.8 percent MAP increase over the previous published results where the parameters were optimized manually. The future efforts will aim towards decreasing the amounts of erroneous trials. One of the solutions is to carefully select the paramaters in the search space to fit the available GPU memory. In order to eliminate the bias towards lower embedding dimensions, separate studies can be constructed for each embedding dimension available.

Acknowledgements. This work has been partly supported by the Czech Science Foundation under the project GA18-23891S. 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. Sabol, R., Medveď, M., Horák, A.: Czech question answering with extended SQAD v3.0 benchmark dataset. (2019) 99–108
2. Medveď, M., Horák, A., Sabol, R.: Improving RNN-based answer selection for morphologically rich languages. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020), Valetta, Malta, SCITEPRESS (2020) 644–651
3. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
4. Patel, D., Raval, P., Parikh, R., Shastri, Y.: Comparative study of machine learning models and BERT on SQuAD (2020)
5. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data (2018)
6. Park, D.H.: Question answering on the squad dataset. (2017)
7. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: In EMNLP. (2014)
8. Tiutiunnyk, S.: Context-based question-answering system for the Ukrainian language. (2020)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12** (2011) 2825–2830
10. 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. (2019) 2623–2631
11. Bergstra, J., Yamins, D., Cox, D.D.: Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms, Citeseer (2013)
12. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 (2018)
13. authors, T.G.: Gpyopt: A bayesian optimization framework in Python. `http://github.com/SheffieldML/GPyOpt` (2016)
14. Probst, P., Wright, M., Boulesteix, A.L.: Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (2018)
15. 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), Setúbal, Portugal, SCITEPRESS - Science and Technology Publications (2018) 486–492
16. Horák, A., Medveď, M.: SQAD: Simple question answering database. In: Eighth Workshop on Recent Advances in Slavonic Natural Language Processing, Brno, Tribun EU (2014) 121–128

17. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for Czech. In: Language and Technology Conference, Springer (2009) 161–171

Part II

Semantics and Language Modelling

Towards Useful Word Embeddings

Evaluation on Information Retrieval, Text Classification, and Language Modeling

Vít Novotný, Michal Štefánik, Dávid Lupták, and Petr Sojka

Faculty of Informatics, Masaryk University
https://mir.fi.muni.cz/

Abstract. Since the seminal work of Mikolov et al. (2013), word vectors of log-bilinear models have found their way into many NLP applications and were extended with the positional model. Although the positional model improves accuracy on the intrinsic English word analogy task, prior work has neglected its evaluation on extrinsic end tasks, which correspond to real-world NLP applications. In this paper, we describe our first steps in evaluating positional weighting on the information retrieval, text classification, and language modeling extrinsic end tasks.

Keywords: Evaluation, word vectors, word2vec, fastText, information retrieval, text classification, language modeling

1 Introduction

In the beginning, there was a word. Word representations produced by log-bilinear models have found their way into many real-world NLP applications such as word similarity, word analogy, and language modeling [2] as well as dependency parsing [8, Section 5], word sense disambiguation [3], text classification [8], semantic text similarity [2], and information retrieval [19, Section 4].

The log-bilinear language of Mikolov et al. (2013) [10] and Bojanowski et al. (2018) [1] have been evaluated on both the intrinsic word analogy tasks and the extrinsic tasks of information retrieval, text classification, and language modeling. By contrast, the positional model of Mikolov et al. (2018) [11] has only been shown to reach SOTA performance on the intrinsic English word analogy task, but its usefulness to real-world NLP applications has been neglected. Our work describes the first steps in evaluating positional weighting on the information retrieval, text classification, and language modeling extrinsic end tasks.

Our work is structured as follows: In Section 2, we will describe the baseline model and the positional model. In Section 3, we will describe the initialization, the parameters, and the datasets for training the models. In Section 4, we will describe real-world applications of word vectors, and the datasets that we used in our experiments. In Section 5, we will show and discuss the results of our experiments. We conclude in Section 6 and suggest directions for future work.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 37–46, 2020.

2 Word embedding models

In this section, we will describe the baseline model and the positional model, which we used in our experiments.

2.1 Baseline model

Continuous bag of words In their seminal work, Mikolov et al. (2013) [10, Section 3.1] have introduced the continuous bag of words (CBOW) model, which is trained to predict a masked word in a context window P from the average $\mathbf{v}_C$ of the context word vectors $\mathbf{d}_p$:

$$\mathbf{v}_C = \frac{1}{|P|} \sum_{p \in P} \mathbf{d}_p. \tag{1}$$

In their later work, Mikolov et al. (2013) [12, Section 2.2] have introduced a faster alternative to the full/hierarchical softmax objectives by using a simplified variant of noise contrastive approximation (NCA) [6] called negative sampling.

Subword model The CBOW model only produces vectors for words that are present in the training corpus. Additionally, no weight sharing is used for different inflectional forms of a word, which slows down training convergence. To address both these points, Bojanowski et al. (2018) [2, Section 3.2] have extended CBOW by modeling subwords instead of words, making the vector $\mathbf{d}_p$ for a context word w_p an average of the vectors $\mathbf{d}_g$ for its subwords $g \in G_{w_p}$.

2.2 Positional model

In many sentences, the position of the context words is important for predicting the masked word. Consider the following two sentences, which produce an identical context vector $\mathbf{v}_C$ despite the different masked words:

1. Unlike dogs, cats are ⟨*mask*⟩.
2. Unlike cats, dogs are ⟨*mask*⟩.

If the context window P is large, distant context words will also be unimportant for predicting the masked word.

To better model these scenarios, Mikolov et al. (2018) [11, Section 2.2] adopted the positional weighting of Mnih and Kavukcuoglu (2013) [13, Section 3]. Positional weighting makes the average $\mathbf{v}_C$ into a weighted average $\mathbf{w}_C$, where the weight of a context word at a position p is the positional vector $\mathbf{u}_p$, and the weighting is carried out using the pointwise vector product $\odot$:

$$\mathbf{w}_C = \frac{1}{|P|} \sum_{p \in P} \mathbf{d}_p \odot \mathbf{u}_p. \tag{2}$$

Mikolov et al. reached SOTA performance on the intrinsic English word analogy task with the positional models. In our work, we will investigate whether positional weighting also improves the performance of the CBOW model on extrinsic end tasks with real-world NLP applications.

3 Experimental setup

In this section, we will describe the initialization, the parameters, and the datasets for training the models, which we used in our word vector experiments.

3.1 Initialization

For our baseline CBOW model, we follow the experimental setup of Bojanowski et al. (2017) [2] and initialize the subword vectors $\mathbf{d}_g$ from the continuous uniform distribution $\mathcal{U}(\pm\frac{1}{D})$, where D is the dimensionality of the subword vectors.

For the positional model, we initialize the subword vectors $\mathbf{d}_g$ to $\mathcal{U}(\pm\frac{1}{D})$ and the positional vectors $\mathbf{u}_p$ to $\mathbf{1}$. This corresponds to the *identity positional vectors* initialization described by Novotný in a separate article from these proceedings.

3.2 Model parameters

For our baseline CBOW model, we follow the experimental setup of Mikolov et al. (2018) [11, Section 4]: $|P| = 5, D = 300$, hash table bucket size $2 \cdot 10^6$, negative sampling loss with 10 negative samples, initial learning rate 0.05 with a linear decay to zero, sampling threshold 10^{-5}, subword sizes $\{3, 4, 5, 6\}$, minimum word count 5, and window size 5.

For the positional model, we follow the experimental setup of Mikolov et al. (2018) [11, Section 4] and increase the context window size to $|P| = 15$. These parameters are the SOTA on the English word analogy task.

3.3 Datasets

We trained both the baseline CBOW model and the positional model on the 2017 English Wikipedia[1] dataset over one epoch. The size of our dataset is only 4% of the Common Crawl dataset used by Mikolov et al. (2018) [11] to reach SOTA performance on the English word analogy task.

4 Applications

In this section, we will describe real-world applications of word vectors, and the datasets that we either used in our experiments or that we plan to use in future.

4.1 Information retrieval

Ad-hoc information retrieval is a standard retrieval task with applications in full-text search engines. In information retrieval, word vectors can be used as a source of word relatedness in semantic text distance or similarity measures such as the word mover's distance (WMD) [8] or the soft cosine measure (SCM) [17].

[1] https://github.com/RaRe-Technologies/gensim-data (release wiki-english-20171001)

We have not yet evaluated word vectors on information retrieval and this section will therefore discuss the Text Retrieval Conferences (TREC) datasets, which have been used for the evaluation of information retrieval systems at TREC conferences, and which we have preprocessed for future experiments.

Introduction The Natural Language Processing Centre at the Faculty of Informatics, Masaryk University (NLP Centre), obtained TREC disks 4 and 5 for research purposes free of charge from the National Institute of Standards and Technology (NIST) in 2019. Together with these test collections, test questions (topics) and relevance judgments are publicly available to evaluate information retrieval systems with the underlying TREC disks 4 and 5 documents. [16]

The Text Research Collection Volume 4 (Disk 4) from May 1996 and the Text Research Collection Volume 5 (Disk 5) from April 1997 were used together in the information retrieval tracks at TREC-6 through 8 in the years 1997–1999. [15] The collections include material from the Financial Times Limited (years 1991–1994), the Federal Register (year 1994), the Foreign Broadcast Information Service (year 1996), and the Los Angeles Times (years 1989 and 1990). [14] All collections consist of English texts, ranging from approximately 55k to 210k documents per collection and with a median of 316 to 588 words per document. [20] For each TREC conference, the organizers provided a set of 50 natural language topic statements [20,21,22] from which participants produced a set of queries that would be run against the collection documents.

The format of both documents and topics is the Standard Generalized Markup Language (SGML), with Document Type Definition (DTD) grammars provided for the documents. These validation capabilities ensure that we have superbly valid documents at our hands, but due to the overall complexity and lenient tagging of SGML, we lack a simple machine-readable format such as the Extensible Markup Language (XML) or the JavaScript Object Notation (JSON). In order to enhance machine-readability, we converted the documents to XML.

Preprocessing As listed above, Disks 4 and 5 consist of four different document collections. Hence, they contain four different DTD files because the dataset creators tended to keep the data as close to the original sources as possible. One of the DTD grammars contains the SGML declaration with parser instructions, which differs from the default Reference Concrete Syntax, mainly in increased limits to provide large document instances, like the quantities, capacities, or lengths of attribute values. None of these limits are restricted in XML. [4] Other changes include e.g. switching the SGML markup minimization features, which are also irrelevant for XML markup. Therefore, our main focus in the DTD files are the declarations of elements, attributes, and entities. These declarations are supported in both SGML DTDs and XML DTDs, which allowed us convert the SGML DTDs to XML DTDs with minimum effort just by following general rules. [9,23]

The inspection of the DTD grammars showed that the structure of the provided data in the SGML format is very close to the generally stricter syntax of XML. This is mainly the case of the syntax for omitting tags, which is present only for the top-level elements that enclose all documents in the collection. One possible

reason is that the data collection itself consists of multiple smaller files for easier manipulation. Otherwise, no markup minimization is present in the data.

To convert element type declarations, we excluded `OMITTAG` specifications since tag minimization is not allowed in XML. Another change was in mixed content models, i.e. elements that contain other elements or text (`#PCDATA`), that must have `#PCDATA` as the first element and must be optional (the `*` operator) in XML. For instance, the `TABLE` element type declaration in SGML DTDs

```
<!ELEMENT TABLE - - (FRFILING* | SIGNER* | #PCDATA)+ >
```

becomes the following in XML DTDs:

```
<!ELEMENT TABLE     (#PCDATA | FRFILING | SIGNER)* >
```

In SGML and XML documents, internal entities specify names, usually for special characters, that we can use in texts for markup, and the entity declaration contains the respective Unicode character reference to which the entity should expand. For most of the entity declarations in the SGML DTDs, we needed to supply numerical character references in decimal or hexadecimal format.

Another significant difference between the SGML and XML syntax is the requirement to put quotes around attribute values in XML, which, on the other hand, can be omitted in SGML. We satisfied this requirement with `osx`, a converter from SGML to XML from the OpenSP library, at the level of document collections.

The final preprocessing steps consisted of joining all chunks of documents into a single file, appending the updated external DTD file, and enclosing the whole data in the top-level element according to the given document collection and DTD grammar. We can validate the data with the `xmllint` XML parser using the following command.

```
$ xmllint --dtdvalid <DTD> <DATA> --noout
```

Conclusion We converted the TREC datasets for information retrieval purposes to a machine-readable format that we can use for our upcoming experiments. The datasets themselves are available[2] to all NLP Centre members who sign the individual agreement for research purposes. Even though the datasets are available under restrictive license conditions, we believe the described preprocessing steps will be useful for broader audiences outside the NLP Centre.

4.2 Text classification

Text classification is an NLP task with applications in automatic categorization and clustering of documents in interactive databases, e-shops, and digital libraries. In text classification, word vectors can be used as a source of word relatedness in semantic text distance or similarity measures such as the word mover's distance (WMD) [8] or the soft cosine measure (SCM) [17], respectively. To enable reproducibility, we publish our experimental code online.[3]

[2] https://nlp.fi.muni.cz/projekty/information-retrieval/
[3] http://drive.google.com/drive/folders/1TtGM0r4etmB8wU7_jtap3zokXd_O2mC2

Experimental setup In our evaluation, we used the WMD [8] and the orthogonalized SCM [17,18] semantic text distance and similarity measures with the k-nearest neighbors (kNN) classifier, following the setup of Kusner et al. (2015) [8]. Unlike Kusner et al., we did not optimize k and we hand-picked the value $k = 11$. For the SCM, we used the default parameters $o = 2, t = 0, C = 100$, Sym = `True`, and Dom = `False` from the implementation[4] of Novotný (2018) [17]. Whereas Kusner et al.'s WMD used the `nnn` BOW scheme for term weighting, our WMD uses the `bnn` binary BOW scheme and the SCM uses the `nfn` TF-IDF scheme.

Datasets As our datasets, we used BBCSPORT, TWITTER, RECIPE, OHSUMED, CLASSIC, REUTERS, and AMAZON from Kusner et al (2015) [8]. In TWITTER, the class labels indicate sentiment, whereas the other datasets are labeled by topic. Due to the high time complexity of the WMD, [18, Section 3] we only evaluated the WMD on the three hardest datasets in terms of the test error reported by Kusner et al. (TWITTER, RECIPE, and OHSUMED). Additionally, whereas Kusner et al. averaged their results on the TWITTER and RECIPE datasets across five train-test splits, we evaluate the WMD using only the first train-test split. To show that this should not significantly affect the accuracy of measurement, we report 95% confidence intervals for the test error of the SCM on the TWITTER and RECIPE datasets. For all datasets, measures, and models, we report test error. For the SCM, we also compute 95% significance on datasets with five splits (all except OHSUMED and REUTERS).

4.3 Language modeling

Language modeling is an NLP task with applications in predictive text, speech recognition, and optical character recognition. In language modeling, word vectors can be used to initialize the lookup table of a recurrent neural network. To enable reproducibility, we publish our experimental code online.[5]

Experimental setup In our evaluation, we trained a single-layer recurrent network similar to Bojanowski et al. (2017) [2, Section 5.6] with the following architecture:

1. an input layer with a map from a vocabulary V of $|V| = 2 \cdot 10^5$ most frequent words to frozen word vectors, followed by
2. an LSTM unit with a recurrent hidden output of size $D = 300$, followed by
3. a fully-connected linear layer of size $|V|$, followed by
4. a softmax output layer that computes probability over the vocabulary V.

We used negative log-likelihood loss, batch size 100, tied weights [7], dropout 0.2, SGD with learning rate 20, and gradient clipping to keep ℓ_2-norm below 0.25.

Datasets As our datasets, we used the data from the 2013 ACL Workshop on Machine Translation[6] with `news-train2011` over five epochs as the training set, `news-commentary-v8` as the validation set, and 10% of `news-train2012` as the test set. We report test perplexity and test loss, and we show learning curves for the training and validation perplexity.

[4] https://radimrehurek.com/gensim/similarities/termsim.html
[5] http://drive.google.com/drive/folders/1fVdNO8ZpfMtdJOWnD3Z1nu78UQPMcItG
[6] https://www.statmt.org/wmt13/translation-task.html

5 Results

In this section, we will show and discuss the results of our experiments on the text classification and language modeling extrinsic end tasks.

5.1 Text classification

Table 1 shows that the positional model outperforms the baseline model on the Bbcsport, Recipe, Ohsumed, Classic, and Amazon text classification datasets and only significantly underperforms the baseline model on the Reuters text classification dataset. This shows that positional weighting produces word vectors that are better for text classification applications than the baseline.

Table 1 also shows that the positional model with the wmd significantly underperforms the baseline model with the wmd on the Twitter sentiment analysis dataset. This is because word vectors capture relatedness and not sentiment. Antonyms such as *good* and *bad* appear in similar sentences, which makes their word vectors similar as well due to the training objective of cbow. [18] Positional weighting makes it easier to satisfy the training objective, which naturally increases the test error on sentiment analysis datasets. This shows that word vectors are generally not well-suited to sentiment analysis applications.

In Table 1, the test error of the wmd with the baseline and positional models was computed using only the first of five train-test splits of the Twitter and Recipe datasets for speed. Since the 95% confidence intervals for the scm are only $\pm$0.81% on Twitter and $\pm$1.19% on Recipe with the baseline model and $\pm$1.03% on Twitter and $\pm$0.97% on Recipe with the positional model, we conclude that the datasets are well-balanced and that the measurements are accurate.

Table 1. Classification error of the baseline and positional models with the wmd and scm measures and the *k*nn classifier on the text classification test sets. For the wmd, we also list the results of Kusner et al. (2015) [8] for comparison. The best results are **emphasized**.

		Bbcsport	Twitter	Recipe	Ohsumed	Classic	Reuters	Amazon
wmd	Kusner [8]	**4.6%**	29%	43%	44%	**2.8%**	**3.5%**	**7.4%**
	Baseline		**23.78%**	43.47%	46.16%			
	Positional		38.20%	**34.23%**	46.32%			
scm	Baseline	6.64%	29.03%	45.63%	41.32%	4.85%	7.58%	10.27%
	Positional	5.82%	28.54%	43.52%	**38.93%**	4.40%	8.73%	9.81%

5.2 Language modeling

Figure 1 shows that the positional model consistently improves the performance during the whole training compared to the baseline model. Figure 1 also shows that our language models have not plateaued and should train for more epochs.

Table 2 shows that the positional model outperforms the baseline on language modeling datasets. This shows that positional weighting produces word vectors that are better for language modeling applications than the baseline.

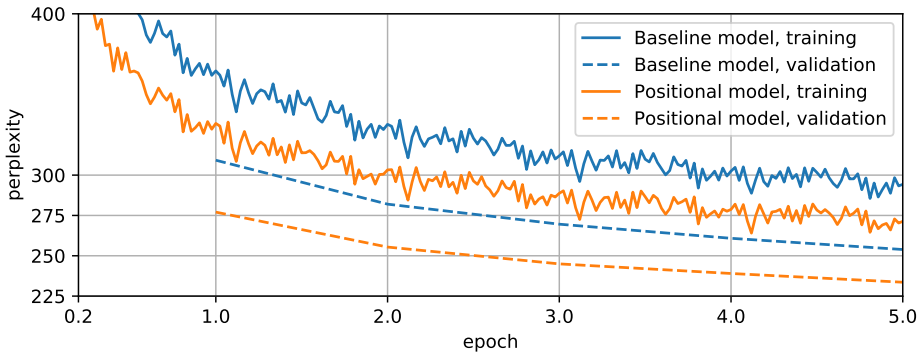


Fig. 1. Learning curves for the perplexity of the baseline and positional models on the language modeling training and validation sets.

Table 2. The perplexity and the loss of the baseline and positional models on the language modeling test set. The best results are **emphasized**.

	Test perplexity	Test loss
Baseline model	270.34	5.60
Positional model	**251.69**	**5.53**

6 Conclusion

We have shown that the positional model produces word vectors that are better-suited to text classification and language modeling NLP applications than the word vectors produced by baseline models. We have also described our steps in preprocessing the TREC information retrieval dataset for word vector evaluation.

In future work, we will evaluate both the positional model and other extensions of the baseline log-bilinear models on information retrieval, dependency parsing, word sense disambiguation, and other extrinsic end tasks using multilingual datasets. We will also train our word vector models using larger corpora such as Common Crawl to enable meaningful comparison to SOTA results.

Acknowledgments. First author's work was funded by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project.

References

1. Bojanowski, P., et al.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
2. Charlet, D., Damnati, G.: Simbow at SemEval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 315–319 (2017)

3. Chen, X., Liu, Z., Sun, M.: A unified model for word sense representation and disambiguation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1025–1035 (2014)
4. Clark, J.: Comparison of SGML and XML. World Wide Web Consortium Note (1997)
5. Clark, K., Khandelwal, U., Levy, O., Manning, C.D.: What does BERT look at? An analysis of BERT's attention. arXiv preprint arXiv:1906.04341 (2019)
6. Gutmann, M.U., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. The Journal of Machine Learning Research, JLMR **13**(1), 307–361 (2012)
7. Inan, H., Khosravi, K., Socher, R.: Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling. CoRR **abs/1611.01462** (2016), http://arxiv.org/abs/1611.01462
8. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From Word Embeddings To Document Distances. In: Bach, F., Blei, D. (eds.) International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 957–966. PMLR, Lille, France (07–09 Jul 2015), http://proceedings.mlr.press/v37/kusnerb15.html
9. McGrath, S.: XML by Example: Building E-Commerce Applications. Prentice Hall PTR (1999)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in Pre-Training Distributed Word Representations. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
12. 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. Curran Associates, Inc. (2013), http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
13. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26, pp. 2265–2273. Curran Associates, Inc. (2013), https://proceedings.neurips.cc/paper/2013/file/db2b4182156b2f1f817860ac9f409ad7-Paper.pdf
14. NIST: Data – English documents (2019), https://trec.nist.gov/data/docs_eng.html
15. NIST: Data – English documents introduction (2019), https://trec.nist.gov/data/intro_eng.html
16. NIST: Test Collections (2019), https://trec.nist.gov/data/test_coll.html
17. Novotný, V.: Implementation Notes for the Soft Cosine Measure. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. p. 1639–1642. CIKM '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3269206.3269317, https://doi.org/10.1145/3269206.3269317
18. Novotný, V., Ayetiran, E.F., Stefánik, M., Sojka, P.: Text classification with word embedding regularization and soft similarity measure. CoRR **abs/2003.05019** (2020), https://arxiv.org/abs/2003.05019
19. Novotný, V., Sojka, P., Štefánik, M., Lupták, D.: Three is better than one. In: CEUR Workshop Proceedings. Thessaloniki, Greece (2020), http://ceur-ws.org/Vol-2696/paper_235.pdf

20. Voorhees, E.M., Harman, D.: Overview of the sixth Text REtrieval Conference (TREC-6). Proceedings of the Sixth Text REtrieval Conference (TREC-6) pp. 1–24 (1998), https://trec.nist.gov/pubs/trec6/t6_proceedings.html, NIST Special Publication 500-240
21. Voorhees, E.M., Harman, D.: Overview of the seventh Text REtrieval Conference (TREC-7). Proceedings of the Seventh Text REtrieval Conference (TREC-7) pp. 1–23 (1999), https://trec.nist.gov/pubs/trec7/t7_proceedings.html, NIST Special Publication 500-242
22. Voorhees, E.M., Harman, D.: Overview of the eighth Text REtrieval Conference (TREC-8). Proceedings of the Eighth Text REtrieval Conference (TREC-8) pp. 1–23 (2000), https://trec.nist.gov/pubs/trec8/t8_proceedings.html, NIST Special Publication 500-246
23. Walsh, N.: Converting an SGML DTD to XML (1998), https://www.xml.com/pub/a/98/07/dtd/index.html

Using FCA for Seeking Relevant Information Sources

Marek Menšík, Adam Albert, and Vojtěch Patschka

Department of Computer Science, FEI,
VŠB - Technical University of Ostrava, 17. listopadu 15, 708 00 Ostrava,
Czech Republic
mensikm@gmail.com, adam.albert@vsb.cz, vojtech.patschka@vsb.cz

Abstract. In this paper, we present using formal concept analysis (FCA) for seeking relevant informational sources from many textual resources. The method is based on explications of an atomic concept formalized as constructions of Transparent Intensional Logic (TIL). In this paper we assume, that all explications have been already done and just shown how FCA can be used as a background of text source recommendation.

1 Introduction

This paper deals with another technique which is possible to use for selecting possibly interesting text sources from a given set of text documents. Whole process is based on applying theory of machine learning and concept explication. Because we needed formalise the sentences in natural languages to some formal language, we decided to use strong system of Transparent intensional logic [1].

In prior papers [2], [3] we introduced methods for selecting relevant text sources. All methods are based o machine learning introduced in [4] and concept explication introduced in [5].

In this paper we also use previous results published in [4] and [5] but the theory of FCA is used for searching other possible relevant text sources. As a comparison with other methods we will present our results by same explications presented in [2] where we use association rules for source recommendation.

In chapter 2 we briefly mention the problem of concept explication which is important for next data processing. In chapter 3 we introduce the theory of Formal concept analysis and *relevant ordering*. Chapter 4 shows the particular example how to apply our method.

2 Explication of an atomic concept

Since we are dealing with natural language processing, we use TIL as our background theory. TIL allows us to formalize salient semantic features of natural language in a fine-grained way. For more details see [1].

By combining TIL and machine learning, we explicate atomic concepts for the purpose of understanding them and for retrieval of additional useful

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 47–54, 2020.

information. *Carnapian explication*[1] is a process of refinement of an ambiguous or vague expression. The expression, to be refined, is called an *explicandum*; its refinement, obtained by the explication, is called an *explicatum*. For example, a simple expression such as a dog (explicandum) can be refined as "*Dog is adomesticated carnivore*" (explicatum). In terms of TIL, the explicandum is an atomic concept, i.e. an atomic closed construction. The explicatum is a molecular construction defining the explicandum. We also say that the molecular concept is an ontological definition of the object falling under the atomic concept.

For example:

$$^0Dog =_{df} \lambda w \lambda t \, \lambda x \, [[^0Domesticated \;\; ^0Carnivore]_{wt} \, x]$$

Types: $Domesticated/((o\iota)_{wt}(o\iota)_{wt}); Dog, Carnivore/(o\iota)_{wt}; x \rightarrow \iota$

Such explication such as one above we obtained by an algorithm we introduced in [2]. The algorithm exploits symbolic method of supervised machine learning adjusted to natural language processing.The input of the algorithm are sentences in natural language mentioning the expression to be explicated formalised as TIL constructions.

The algorithm, based on Patrick Winston's work [7], iteratively builds the explicatum using the constructions marked as positive or negative examples. With positive examples, we refine the explicatum by inserting new constituents into molecular the construction or we generalize the expliactum so it can adequately define the explicandum. With negative exmples, we specialize the explicatum by inserting new constituents in negated way. By those constituents we differentiate the expliacum of our expression from similar expression's explicatum.

3 FCA and relevant ordering

Formal Conceptual Analysis[2] (FCA) was introduced in 1980s by group lead by Rudolf Wille and became a popular technique within the information retrieval field. 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.

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. ***Significant objects*** *of object* ***e*** *in* $\beta(G, M, I)$ *is set* $SO(e) = \bigcup_{i=1}^{n} O_i^e$, *where* O^e *is extent of a concept* $(O, A) \neq (G, B), e \in O, B \subseteq M$. *Namely, significant objects of object* ***e*** *is union of all extents where the object* ***e*** *is as an element.*

[1] See [6]

[2] More in [8].

Definition 3. *Let $SO(e)$ is a set of significant objects of an object* ***e****, let $\gamma(e)$ is a set of concepts (O, A) where $e \in O$,i.e.: $\gamma(e) = \{(O^e, (O^e)^\uparrow) | (O^e, (O^e)^\uparrow) \neq (G, B), B \subseteq M, (O^e, (O^e)^\uparrow) \in \beta(G, M, I)\}$, then* $\boldsymbol{a \sqsubseteq b}$ *is in* ***relevant ordering***[3] *iff $max(|(O^a)^\uparrow|) \leq max(|(O^b)^\uparrow|), a, b, \in SO(e), (O^a, (O^a)^\uparrow), (O^b, (O^b)^\uparrow) \in \gamma(e)$.*

Example: Let have a formal context described by the following Table 1.

Table 1. Formal context

O/A	a_0	a_1	a_2	a_3
o_0	1	1	0	0
o_1	0	1	1	0
o_2	0	0	1	1

The set of all *formal concepts* $\beta(G, M, I) = \{C_0, C_1, C_2, C_3, C_4, C_5, C_6, \}$, where $C_0 = (\{o_0, o_1, o_2\}, \emptyset), C_1 = (\{o_0, o_1\}, \{a_1\}), C_2 = (\{o_0\}, \{a_0, a_1\}), C_3 = (\{o_1, o_2\}, \{a_2\}), C_4 = (\{o_1\}, \{a_1, a_2\}), C_5 = (\{o_2\}, \{a_2, a_3\}), C_6 = (\emptyset, \{a_0, a_1, a_2, a_3\})$

Get the set of *significant objects* of object o_2, $SO(o_2)$:

1. Find set $\gamma(o_2) \rightarrow \gamma(o_2) = \{(\{o_1, o_2\}, \{a_2\}), (\{o_2\}, \{a_2, a_3\})\}$
2. Find all *extents* O_i from $\beta(G, M, I)$ where $O_i \neq G$ and $o_2 \in O_i \rightarrow O_1^{o_2} = \{o_2\}, O_2^{o_2} = \{o_1, o_2\}$
3. Create the union of all extents found in step 2 $\rightarrow SO(o_2) = \{o_1, o_2\}$

Find *relevant ordering* of $SO(o_2)$

1. For all $x \in SO(o_2)$ calculate max of $|(O^x)^\uparrow|$, where $((O^x), (O^x)^\uparrow) \in \gamma(o_2) \rightarrow max(|(O^{o_1})^\uparrow|) = 1, max(|(O^{o_2})^\uparrow|) = 2$
2. Order $SO(o_2)$ by definition 3 $\rightarrow o_1 \sqsubseteq o_2$

Table 2. Relevant object ordering

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

In our table the column DF represents the difference from the selected object (first row in table - o_2). The column RT will represent the document which is represented by the particular object in column Exp.

Remark: In our study, every object represents one explication of a particular natural language concept. From that point of view, the set of all constituents (row in a table) represents *intent* of a particular formal concept. There exist a formal concept $(\{e\}, \{e\}^\uparrow)$ for each explication e.

[3] Classical concept ordering is defined as: $(O, A) \sqsubseteq (O_1, A_1)$ iff $A \subseteq A_1$

4 Demonstration on example

As an example of recommending relevant information sources based on FCA, we use the same data we used in [2]. In our example we used text sources dealing with a concept *wild cat*. We obtained 8 explicates of the concept from different textual sources ($s_1, ..., s_8$). That means that each explication describes the concept of *being a wild cat* from other point of view. Those explications are following:

$e_1 = [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['\leq ['Weight_{wt}\ x]\ '11]\ \wedge\ ['\geq ['Weight_{wt}\ x]\ '1.2]]['Wild\ 'Cat]]\ \wedge$
$['Req\ 'Mammal\ ['Wild\ 'Cat]]\ \wedge\ ['Req\ 'Has\text{-}fur\ ['Wild\ 'Cat]]\ \wedge [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['\leq$
$[['Average\ 'Body\text{-}Length]\ x]\ '80]\ \wedge\ ['\geq [['Average\ 'Body\text{-}Length]\ x]\ '47]]['Wild\ 'Cat]] \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['= [['Average\ 'Skull\text{-}Size]\ x]\ '41.25]]['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['=$
$[['Average\ 'Height]\ x]\ '37,6]]['Wild\ 'Cat]]$

$e_2 = [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['Live\text{-}in_{wt}\ [\lambda w \lambda t \lambda y[[['Mixed\ 'Forrest]_{wt}\ y]\ \vee$
$[['Deciduous\ 'Forrest]_{wt}\ y]]]]['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\geq$
$['Territory\text{-}Size_{wt}\ x]\ '50]['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing]\ \vee$
$['Ter\text{-}Marking_{wt}\ x\ 'Urinating]\ \vee\ [Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]['Wild\ 'Cat]]$

$e_3 = [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['\leq ['In\text{-}Heat\text{-}Periodwt\ x]\ '8]\ \wedge\ ['\geq$
$['In\text{-}Heat\text{-}Periodwt\ x]\ '2]]\ ['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x['Seek_{wt}\ x\ 'Mate\ ['Loud\ 'Meow]]\ ['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['=$
$['Pregnancy\text{-}Periodwt\ x]\ '65]\ ['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['\leq$
$['Litter\text{-}Size_{wt}\ x]\ '4]\ \wedge\ ['\geq ['Litter\text{-}Size_{wt}\ x]\ '3]]\ ['Wild\ 'Cat]]$

$e_4 = ['Req\ 'Mammal\ ['Wild\ 'Cat]]\ \wedge\ ['Req\ 'Has\text{-}fur\ ['Wild\ 'Cat]]\ \wedge [Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['=$
$[['Average\ 'Skull\text{-}Size]\ x]\ '41.25]]['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing]\ \vee\ ['Ter\text{-}Marking_{wt}\ x\ 'Urinating]\ \vee$
$[Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['=$
$['Pregnancy\text{-}Periodwt\ x]\ '65]\ ['Wild\ 'Cat]]\ \wedge [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\leq$
$['Litter\text{-}Size_{wt}\ x]\ '4]\ ['Wild\ 'Cat]]$

$e_5 = [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\geq [['Average\ 'Body\text{-}Length]\ x]\ '47]['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing]\ \vee\ ['Ter\text{-}Marking_{wt}\ x\ 'Urinating]\ \vee$
$[Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['=$
$['Pregnancy\text{-}Periodwt\ x]\ '65]\ ['Wild\ 'Cat]]\ \wedge [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\leq$
$['Litter\text{-}Size_{wt}\ x]\ '4]\ ['Wild\ 'Cat]]$

$e_6 = [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\geq [['Average\ 'Body\text{-}Length]\ x]\ '47]['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing]\ \vee\ ['Ter\text{-}Marking_{wt}\ x\ 'Urinating]\ \vee$
$[Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x['Seek_{wt}\ x\ 'Mate\ ['Loud\ 'Meow]]\ ['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\leq$
$['Litter\text{-}Size_{wt}\ x]\ '4]\ ['Wild\ 'Cat]]$

$e_7 = ['Req\ 'Mammal\ ['Wild\ 'Cat]]\ \wedge\ ['Req\ 'Has\text{-}fur\ ['Wild\ 'Cat]]\ \wedge\ [Typ\text{-}p\ \lambda w \lambda t\ \lambda x['\leq$
$['Weight_{wt}\ x]\ '11]\ ['Wild\ 'Cat]]\ \wedge$
$[Typ\text{-}p\ \lambda w \lambda t\ \lambda x['Live\text{-}in_{wt}\ [\lambda w \lambda t \lambda y[[['Mixed\ 'Forrest]_{wt}\ y]\ \vee$

$[['Deciduous\ 'Forrest]_{wt}\ y]]]]['Wild\ 'Cat]] \wedge ['Typ\text{-}p\ \lambda w\lambda t\ \lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing] \vee ['Ter\text{-}Marking_{wt}\ x\ 'Urinating] \vee ['Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]['Wild\ 'Cat]] \wedge ['Typ\text{-}p\ \lambda w\lambda t\ \lambda x['Seek_{wt}\ x\ 'Mate\ ['Loud\ 'Meow]]\ ['Wild\ 'Cat]] \wedge ['Typ\text{-}p\ \lambda w\lambda t\ \lambda x['= ['Pregnancy\text{-}Periodwt\ x]\ '65]\ ['Wild\ 'Cat]]$

$e_8 = ['Typ\text{-}p\ \lambda w\lambda t\ \lambda x['\geq [['Average\ 'Body\text{-}Length]\ x]\ '47]\ ['Wild\ 'Cat]] \wedge [\lambda w\lambda t\lambda x['\geq ['Territory\text{-}Size_{wt}\ x]\ '50]['Wild\ 'Cat]] \wedge ['Typ\text{-}p\ \lambda w\lambda t\ \lambda x['\leq ['Litter\text{-}Size_{wt}\ x]\ '4]\ ['Wild\ 'Cat]]$

After obtaining all explications the user selects one of them which is the most relevant from his point of view. In our case e_1. The whole process of recommendation starts after the explication selection.

From explications mentioned above, we make an incidence matrix written in Table 3.

Each row represents one explication and each column represents particular property. The value 1 represent that the explication contains the property, value 0 represents that the explication doesn't have the property.

The $e_1, ..., e_8$ are identifiers of the explications.

Table 3. Formal context of explications

O/A	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
$\mathbf{e_1}$	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
$\mathbf{e_2}$	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
$\mathbf{e_3}$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
$\mathbf{e_4}$	1	1	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0
$\mathbf{e_5}$	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0
$\mathbf{e_6}$	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0
$\mathbf{e_7}$	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0
$\mathbf{e_8}$	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0

The columns' numbers in Table 3 represent the following attributes:

1. $'Mammal$
2. $'Has-fur$
3. $\lambda w\lambda t\lambda x['\leq ['Weight_{wt}\ x]\ '11]$
4. $\lambda w\lambda t\lambda x['\geq ['Weight_{wt}\ x]\ '1.2]$
5. $\lambda w\lambda t\lambda x['\geq [['Average\ 'Body\text{-}Length]\ x]\ '47]$
6. $\lambda w\lambda t\lambda x['\leq [['Average\ 'Body\text{-}Length]\ x]\ '80]$
7. $\lambda w\lambda t\lambda x['= [['Average\ 'Skull\text{-}Size]\ x]\ '41.25]$
8. $\lambda w\lambda t\lambda x['= [['Average\ 'Skull\text{-}Height]\ x]\ '37.6]$
9. $\lambda w\lambda t\lambda x\ ['Live\text{-}in_{wt}\ [\lambda w\lambda t\lambda y[[['Mixed\ 'Forrest]_{wt}\ y] \vee [['Deciduous\ 'Forrest]_{wt}\ y]]]]$
10. $\lambda w\lambda t\lambda x['\geq ['Territory\text{-}Size_{wt}\ x]\ '50]$
11. $\lambda w\lambda t\lambda x[['Ter\text{-}Marking_{wt}\ x\ 'Clawing] \vee ['Ter\text{-}Marking_{wt}\ x\ 'Urinating] \vee ['Ter\text{-}Marking_{wt}\ x\ 'Leaves\text{-}Droppings]]$
12. $\lambda w\lambda t\lambda x['\leq ['In\text{-}Heat\text{-}Periodwt\ x]\ '8]$
13. $\lambda w\lambda t\lambda x['\geq ['In\text{-}Heat\text{-}Periodwt\ x]\ '2]$
14. $\lambda w\lambda t\lambda x['Seek_{wt}\ x\ 'Mate\ ['Loud\ 'Meow]]$
15. $\lambda w\lambda t\lambda x['= ['Pregnancy\text{-}Periodwt\ x]\ '65]$
16. $\lambda w\lambda t\lambda x['\leq ['Litter\text{-}Size_{wt}\ x]\ '4]$
17. $\lambda w\lambda t\lambda x['\geq ['Litter\text{-}Size_{wt}\ x]\ '3]$

From Table 3, we obtained following concepts by using FCA:

0. $(\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}, \emptyset)$
1. $(\{e_1, e_4, e_7\}, \{1, 2\})$
2. $(\{e_1, e_4\}, \{1, 2, 7\})$
3. $(\{e_1, e_5, e_6, e_8\}, \{5\})$
4. $(\{e_1, e_7\}, \{1, 2, 3\})$
5. $(\{e_1\}, \{1, 2, 3, 4, 5, 6, 7, 8\})$
6. $(\{e_2, e_4, e_5, e_6, e_7\}, \{11\})$
7. $(\{e_2, e_7\}, \{9, 11\})$
8. $(\{e_2, e_8\}, \{10\})$
9. $(\{e_2\}, \{9, 10, 11\})$
10. $(\{e_3, e_4, e_5, e_6, e_8\}, \{16\})$
11. $(\{e_3, e_4, e_5, e_7\}, \{15\})$
12. $(\{e_3, e_4, e_5\}, \{15, 16\})$
13. $(\{e_3, e_6, e_7\}, \{14\})$
14. $(\{e_3, e_6\}, \{14, 16\})$
15. $(\{e_3, e_7\}, \{14, 15\})$
16. $(\{e_3\}, \{12, 13, 14, 15, 16, 17\})$
17. $(\{e_4, e_5, e_6\}, \{11, 16\})$
18. $(\{e_4, e_5, e_7\}, \{11, 15\})$
19. $(\{e_4, e_5\}, \{11, 15, 16\})$
20. $(\{e_4, e_7\}, \{1, 2, 11, 15\})$
21. $(\{e_4\}, \{1, 2, 7, 11, 15, 16\})$
22. $(\{e_5, e_6, e_8\}, \{5, 16\})$
23. $(\{e_5, e_6\}, \{5, 11, 16\})$
24. $(\{e_5\}, \{5, 11, 15, 16\})$
25. $(\{e_6, e_7\}, \{11, 14\})$
26. $(\{e_6\}, \{5, 11, 14, 16\})$
27. $(\{e_7\}, \{1, 2, 3, 9, 11, 14, 15\})$
28. $(\{e_8\}, \{5, 10, 16\})$
29. $(\emptyset, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\})$

Conceptual lattice of obtained formal concepts is visualised in Fig. 1. Dark nodes represent concepts which *extent* contains only *significant objects*. The nodes with bright numbers represent the particular explications.

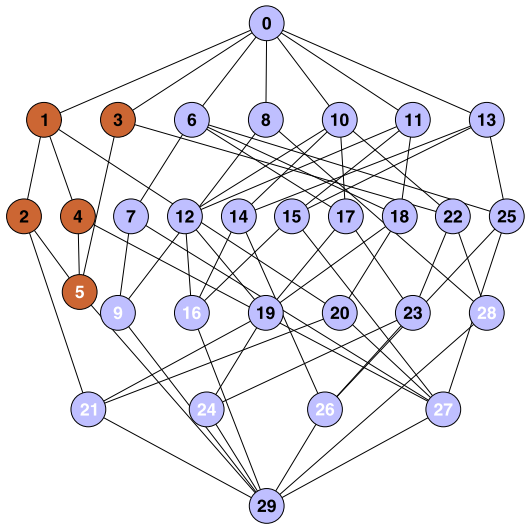


Fig. 1. Lattice of formal concepts

Significant objects of object (explication) e_1 is following set: $SO(e_1) = \{e_1, e_4, e_5, e_6, e_7, e_8\}$. The set of all concepts which have our explication e_1 as a mutual object is the following set:

$$\gamma(e_1) = \{(\{e_1\}, \{1,2,3,4,5,6,7,8\}), (\{e_1, e_4\}, \{1,2,7\}), (\{e_1, e_7\}, \{1,2,3\}), (\{e_1, e_5, e_6, e_8\}, \{5\}), (\{e_1, e_4, e_7\}, \{1,2\}), \}$$

The ordering is represented by following Table 4. The higher the row is the higher priority (relevance) the document (text source) has.

It is clear that the first row represents the document of selected explication (in our case e_1) and the next rows represents documents which has explications obtaining the largest mutual intent with descending tendency.

Table 4. Final text sources' ordering

Exp.	Intent	DF	RT
e_1	{1,2,3,4,5,6,7,8}	{}	$\{s_1\}$
e_4	{1,2,7}	{3,4,5,6,8}	$\{s_4\}$
e_7	{1,2,3}	{4,5,6,7,8}	$\{s_7\}$
e_5	{5}	{1,2,3,4,6,7,8}	$\{s_5\}$
e_6	{5}	{1,2,3,4,6,7,8}	$\{s_6\}$
e_8	{5}	{1,2,3,4,6,7,8}	$\{s_8\}$

Explicitly the relevant text sources ordering is as follows:

$$e_8(s_8) \sqsubseteq e_6(s_6) \sqsubseteq e_5(s_5) \sqsubseteq e_7(s_7) \sqsubseteq e_4(s_4) \sqsubseteq e_1(s_1)$$

5 Conclusion

In this paper, we introduced method which uses the FCA for selecting the most relevant text sources and we introduced *relevant ordering* to order the set of selected explications from the most relevant to the less ones. The goal was to introduce method which could help the user to organize text sources from the most significant therefore the user does not need to go through all documents to get the relevant information.

We are aware of the time consuming method of FCA. In the future we will focus on some modifications which will strongly reduce the time of data post-processing.

Acknowledgments. This research has been supported by the Grant Agency of the Czech Republic, project No. GA18-23891S, "Hyperintensional Reasoning over Natural Language Texts," and by Grant of SGS No. SP2020/62, VŠB – Technical University of Ostrava, Czech Republic.

References

1. Duží, M., Jespersen, B., Materna, P. (2010). Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic. Berlin: Springer.
2. Albert, A., Duží, M., Menšík, M., Pajr, M., Patschka, V. (2020): Search for Appropriate Textual Information Sources. In the proceedings of EJC 2020, 30th International Conference on Informational Modelling and Knowledge Bases, Bernhard Thalheim, Marina Tropmann-Frick, Hannu Jaakkola & Yasushi Kiyoki (eds.), June 8-9, 2020, Hamburg, Germany, pp. 228-247.
3. Menšík, M., Duží, M., Albert, A., Patschka, V., Pajr, M. (2019): Seeking relevant information sources. In Informatics'2019, IEEE 15th International Scientific Conference on In-formatics, Poprad, Slovakia, pp. 271-276.
4. Menšík, M., Duží, M., Albert, A. Patschka, V., Pajr, M. (2020): Machine Learning using TIL. In Frontiers in Artificial Intelligence and Applications, vol. 321: Information Modelling and Knowledge Bases XXXI, A. Dahanayake, J. Huiskonen, Y. Kiyoki, B. Thalheim, H. Jaakkola, N. Yoshida (eds.), pp. 344-362, Amsterdam: IOS Press.
5. Menšík, M., Duží, M., Albert, A., Patschka, V., Pajr, M. (2019): Refining concepts by ma-chine learning. Computación y Sistemas, Vol. 23, No. 3, 2019, pp. 943–958, doi: 10.13053/CyS-23-3-3242
6. Carnap, Rudolf (1964). Meaning and Necessity: A Study in Semantics and Modal Logic. Chicago: University of Chicago Press.
7. Winston P. H.(1992): Artificial intelligence. 3rd ed., Mass.: Addison-Wesley Pub. Co., 1992.
8. Ganter, B., Wille, R. (1999): Formal Concept Analysis: Mathematical Foundations. 1st ed., Berlin: Springer. ISBN 978-3-540-62771-5.

The Art of Reproducible Machine Learning
A Survey of Methodology in Word Vector Experiments

Vít Novotný

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

Abstract. Since the seminal work of Mikolov et al. (2013), word vectors of log-bilinear SVMS have found their way into many NLP applications as an unsupervised measure of word relatedness. Due to the rapid pace of research and the publish-or-perish mantra of academic publishing, word vector experiments contain undisclosed parameters, which make them difficult to reproduce. In our work, we introduce the experiments and their parameters, compare the published experimental results with our own, and suggest default parameter settings and ways to make previous and future experiments easier to reproduce. We show that the lack of variable control can cause up to 24% difference in accuracy on the word analogy tasks.

Keywords: Machine learning, word vectors, word2vec, fastText, word analogy, reproducibility

1 Introduction

After a long reign of topic modeling, [9] log-bilinear SVMS have emerged as a faster[1] method for learning word representations, [18] which can also infer representations of unseen words using a subword model [2]. Word vectors produced by log-bilinear SVMS have found their way into many NLP applications, such as word similarity, word analogy, and language modeling [2] as well as dependency parsing [8, Section 5], word sense disambiguation [6], text classification [14], semantic text similarity [5], and information retrieval [22, Section 4].

Although the usefulness of word vectors is rarely disputed, their theoretical foundations were only later addressed by Levy and Goldberg (2014) [16]. In their later work, Levy at al. (2015) [17] have shown that several pre-processing steps and fixed parameters can have a significant impact on experimental results. In this work, we describe six new undisclosed parameters that make word vector experiments difficult to reproduce. We compare the published experimental results with our own and suggest improvements to reproducibility.

[1] The time complexity of streaming approximations of the sparse SVD method used in topic modeling scales quadratically with the word vector size, [31, Section 2.3] whereas log-bilinear SVMS are linear in both the word vector size and the corpus size.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 55–64, 2020.

Table 1. The accuracies (%) of word vectors on the word analogy tasks for various languages using only the n most frequent words as the word analogy candidates for different values of n. Three languages most affected by the variations of n are *highlighted*.

	CS	DE	ES	*FI*	FR	*HI*	IT	*PL*	PT	ZH
Grave et al. [10], $n = 2 \cdot 10^5$	69.9	72.9	65.4	*70.3*	73.6	*32.1*	69.8	*67.9*	66.7	78.4
Our results, $n = 2 \cdot 10^5$	70.7	73.4	65.6	*71.2*	73.7	*32.2*	73.0	*68.5*	67.0	78.5
Our results, $n = 3 \cdot 10^5$	68.9	73.3	65.6	*68.9*	73.2	*26.4*	72.0	*66.3*	65.7	78.5
Our results, $n = 1 \cdot 10^6$	65.4	70.4	63.4	*60.9*	71.9	*16.0*	68.3	*61.1*	61.3	78.3

2 Word Analogy Tasks

In their seminal work, Mikolov et al. (2013) [18, Section 4] introduced the English word analogy task, which measures the ability of word vectors to answer the question "Which word b' is to a' as a is to b?". In the following years, the English word analogy task has been translated to many languages, including Czech [28], German [15], Spanish [4], Finnish [30], Italian [1], Portuguese [12], Turkish [27,11] and simplified Chinese [7] as well as French, Hindi, and Polish [2].

Rogers et al. (2017) [26] discuss the many problems with the word analogy task, including the selection of word pairs, the significant impact of including/excluding the words a, b, and a' in the candidates for b', and the underlying assumption that word relations are unique or even symmetric. In this section, we discuss two undisclosed parameters of the word analogy task.

2.1 Using only the most frequent words

To make the evaluation less susceptible to rare words, Mikolov et al. (2013) [18] only considered the n most frequent words as the candidates for b'. However, the value of n changes between experiments and is usually undocumented: Mikolov et al. (2013) [18] used $n = 1 \cdot 10^6$, the reference implementation[2] defaults to $n = 3 \cdot 10^5$, and Grave et al. (2018) [10] used $n = 2 \cdot 10^5$. Mikolov et al. (2013) [20], Bojanowski et al. (2017) [2], and Mikolov et al. (2018) [19] did not disclose the value n they used, which makes their results difficult to reproduce.

To show how significant the value of n is, we reproduce[3] the results of Grave et al. (2018) [10, Table 4] with $n \in \{2 \cdot 10^5, 3 \cdot 10^5, 1 \cdot 10^6\}$. Table 1 shows that up to 16% of word analogy accuracy can depend on the value of n. The most affected languages are Hindi, French, and Polish, indicating small/noisy training data.

To make results reproducible, we suggest that all papers should report the value of n they used in their evaluation on the word analogy tasks. If unreported, the value should be assumed to be $3 \cdot 10^5$, which is the default in the reference implementation and in a popular implementation from the Gensim library[4] [25].

[2] https://github.com/tmikolov/word2vec (file compute-accuracy.c)
[3] https://github.com/mir-mu/reproducible-ml (file word-analogy.ipynb)
[4] https://github.com/rare-technologies/gensim (file gensim/models/keyedvectors.py, method evaluate_word_analogies, also discussed in issue #2999)

Table 2. The accuracies (%) of word vectors on the word analogy tasks for various languages and case transformations (upper-casing and lower-casing) with either the U.S. English locale or the corresponding locales for the word analogy task languages. Three languages most affected by the variations are *highlighted*.

	CS	DE	ES	*FI*	*FR*	HI	IT	PL	PT	*TR*	ZH
Grave et al. [10]	69.9	72.9	65.4	*70.3*	*73.6*	32.1	69.8	67.9	66.7		78.4
Our res., no case tran.	69.9	74.9	63.9	*53.3*	*76.7*	32.2	71.9	71.4	67.5	*58.2*	78.5
Our res., u.c., U.S. En.	70.7	73.4	65.6	*71.2*	*73.7*	32.2	73.0	68.5	67.0	*57.0*	78.5
Our res., u.c., corres.	70.7	73.4	65.6	*71.2*	*73.7*	32.2	73.0	68.5	67.0	*61.0*	78.5
Our res., l.c., U.S. En.	70.7	73.4	65.6	*71.2*	*73.7*	32.2	73.0	68.5	67.0	*56.9*	78.5
Our res., l.c., corres.	70.7	73.4	65.6	*71.2*	*73.7*	32.2	73.0	68.5	67.0	*61.0*	78.5

2.2 Caseless matching

To make the evaluation less susceptible to case, the words a, b, a', and b' are all lower-cased in the experiments of Bojanowski et al. (2017) [2]. This is never mentioned in the published papers, only in the code of the reference implementation.[5][6] Another problem is that in Unicode, lower-casing is locale-sensitive:

1. Lower-casing maps I to ı in Turkish and Azari, and to i in other locales.

A popular implementation in Gensim[7] [25] uses upper-casing instead of lower-casing. However, Unicode case is neither bijective nor transitive:

2. Upper-casing maps ß to SS, and lower-casing maps SS to ss (not ß).

This introduces several uncontrolled variables to the evaluation, most importantly the locale and the case transformation used for caseless matching.

To show how significant the locale and the case transformation are, we reproduce[8] the results of Grave et al. (2018) [10, Table 4] with various case transformations, using either the U.S. English locale (`en_US.UTF-8`), or the corresponding locales of the word analogy tasks. Table 2 shows that up to 18% of word analogy accuracy can depend on the case transformations and the locale. The most affected languages are Finnish, Turkish, and French.

To make results reproducible, we suggest that all papers should report the case transformations and locales they used in their evaluation on the word analogy tasks. If unreported, the locale of the word analogy task should be assumed. For case transformation, we suggest using the locale-independent Unicode case-folding algorithm [29, Section 3.13] instead of lower- or upper-casing:

3. Case-folding maps I to i in all locales, although implementations such as ICU can map[9] I to ı for Turkish and Azari. Case-folding maps ß, SS, and ss to ss.

[5] https://github.com/facebookresearch/fastText (file get-wikimedia.sh)
[6] https://github.com/facebookresearch/fastText/blob/master/python/doc/examples/compute_accuracy.py (function process_question)
[7] https://github.com/rare-technologies/gensim (file gensim/models/keyedvectors.py, method evaluate_word_analogies, also discussed in issue #2999)
[8] https://github.com/mir-mu/reproducible-ml (file word-analogy.ipynb)
[9] https://github.com/unicode-org/icu (file icu4c/source/common/unistr_case.cpp, method UnicodeString::foldCase)

3 Multi-Word Expressions

In their work, Mikolov et al. (2013) [20, Section 4] introduced a phrasing algorithm for merging commonly co-occuring words into multi-word expressions. The algorithm forms phrases using unigram and bigram counts, using the following scoring formula, which is proportional to the non-normalized pointwise mutual information (NPMI) [3]:

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j)}{\text{count}(w_i) \cdot \text{count}(w_j)}. \tag{1}$$

Mikolov et al. (2013) merged candidate bigrams $w_i w_j$ with $\text{score}(w_i, w_j)$ above a threshold δ into phrases. Mikolov et al. (2018) [19, Section 2.3] further improved the algorithm by randomly merging only 50% of the candidate bigrams, and reached SOTA performance on the English word analogy task.

In this section, we discuss three undisclosed parameters of the phrasing algorithm and the differences between the reference implementation[10] and a popular implementation in Gensim[11] [25].

3.1 Thresholding the bigram scores

Neither Mikolov et al. (2013) nor Mikolov et al. (2018) disclosed the threshold δ they used for merging candidate bigrams into phrases, which makes their results difficult to reproduce. The reference implementation uses $\delta_{\text{reference}} = 100$ and a different formula:

$$\text{score}_{\text{reference}}(w_i, w_j) = \frac{\text{count}(w_i w_j) \cdot \text{corpusSize}}{\text{count}(w_i) \cdot \text{count}(w_j)} \tag{2}$$

The implementation in Gensim uses $\delta_{\text{Gensim}} = 10$ and also a different formula:

$$\text{score}_{\text{Gensim}}(w_i, w_j) = \frac{\text{count}(w_i w_j) \cdot \text{dictionarySize}}{\text{count}(w_i) \cdot \text{count}(w_j)} \tag{3}$$

Apparently, $\text{score} \neq \text{score}_{\text{reference}} \neq \text{score}_{\text{Gensim}}$. Due to the Heaps' law [13], $\text{dictionarySize} \approx \sqrt{\text{corpusSize}}$, so we might assume $\text{score}_{\text{Gensim}} \approx \sqrt{\text{score}_{\text{ref.}}}$. Since $\delta_{\text{Gensim}} = \sqrt{\delta_{\text{reference}}}$, we might then conclude $\text{score}_{\text{Gensim}} > \delta_{\text{Gensim}} \iff \sqrt{\text{score}_{\text{reference}}} > \sqrt{\delta_{\text{reference}}}$. However, the assumption does not actually hold:

$$\text{score}_{\text{Gensim}} \approx \frac{\text{count}(w_i w_j) \cdot \sqrt{\text{corpusSize}}}{\text{count}(w_i) \cdot \text{count}(w_j)} \neq \sqrt{\frac{\text{count}(w_i w_j) \cdot \text{corpusSize}}{\text{count}(w_i) \cdot \text{count}(w_j)}}. \tag{4}$$

To make results reproducible, we suggest that all papers should report the scoring formula and the value of δ they used for phrasing. If unreported, the $\text{score}_{\text{reference}}$ scoring formula and the $\delta_{\text{reference}} = 100$ value should be assumed.

[10] https://github.com/tmikolov/word2vec (file word2phrase.c)

[11] https://github.com/rare-technologies/gensim (file gensim/models/phrases.py, class Phrases and function bigram_scorer)

3.2 Incremental threshold decay

Mikolov et al. (2013) [20, Section 4] and Mikolov et al. (2018) [19, Section 2.2] apply the phrasing algorithm iteratively to form longer multi-word expressions. Mikolov et al. (2013) use 2–4 iterations, whereas Mikolov et al. (2018) use 5–6 iterations. Mikolov et al. (2013) also reports using a decaying threshold to make it easier for longer phrases to form. However, neither Mikolov et al. (2013) nor Mikolov et al. (2018) disclosed the threshold decay function they used, which makes their results difficult to reproduce.

To make results reproducible, we suggest that all papers should report the exact number of iterations and the threshold decay function they used.

3.3 Maximum dictionary size

To make the phrasing algorithm less susceptible to rare words, Mikolov et al. (2013) [20] and Mikolov et al. (2018) [19] have only considered the n most frequent words for the candidate bigrams. This is mentioned only in the code of the reference implementation. Additionally, the reference implementation uses $n = 5 \cdot 10^8$, whereas a popular implementation in Gensim uses $n = 4 \cdot 10^7$.

To make results reproducible, we suggest that all papers should report the value of n they used for phrasing. If unreported, the value should be assumed to be $n = 5 \cdot 10^8$, which is the default in the reference implementation.

4 Positional Weighting

The CBOW model of Mikolov et al. (2013) [18] is trained to predict a masked word in a context window P from the average $\mathbf{v}_C$ of the context word vectors $\mathbf{d}_p$:

$$\mathbf{v}_C = \frac{1}{|P|} \sum_{p \in P} \mathbf{d}_p. \tag{5}$$

In many sentences, the position of the context words is important for predicting the masked word. Consider the following two sentences, which produce an identical context vector $\mathbf{v}_C$, although the masked words are significantly different:

1. Unlike dogs, cats are ⟨*mask*⟩.
2. Unlike cats, dogs are ⟨*mask*⟩.

If the context window P is large, distant context words will also be unimportant for predicting the masked word.

To better model these scenarios, Mikolov et al. (2018) [19, Section 2.2] adopted the positional weighting of Mnih and Kavukcuoglu (2013) [21, Section 3], and reached SOTA performance on the English word analogy task. Positional weighting makes the average $\mathbf{v}_C$ into a weighted average $\mathbf{w}_C$, where the weight of a context word at a position p is the positional vector $\mathbf{u}_p$, and the weighting is carried out using the pointwise (Hadamard) vector product $\odot$:

$$\mathbf{w}_C = \frac{1}{|P|} \sum_{p \in P} \mathbf{d}_p \odot \mathbf{u}_p, \tag{6}$$

In this section, we discuss an undisclosed parameter of positional weighting.

4.1 Positional weight initialization

In the CBOW model of Mikolov et al. (2013) [18], the word vectors $\mathbf{d}_p$ are initialized to a random sample of the continuous uniform distribution $\mathcal{U}(\pm\frac{1}{2D})$, where D is the dimensionality of the word vectors. Word vector initialization is an important parameter that affects the gradient size and therefore the effective learning rate. However, it is never mentioned in the published papers, only in the code of the reference implementation.[12] In the subword CBOW model of Bojanowski et al. (2017) [2], the initialization changes to $\mathbf{d}_p \sim \mathcal{U}(\pm\frac{1}{D})$ in the code of the reference implementation.[13] Mikolov et al. (2018) [19] do not describe the initialization of the word vectors $\mathbf{d}_p$ or the positional vectors $\mathbf{u}_p$. Since no reference implementation exists either, their results are difficult to reproduce.

To show how significant initialization is, we describe several initialization options for the word vectors $\mathbf{d}_p$ and the positional vectors $\mathbf{u}_p$. We then use the initializations to reproduce[14] the results of Mikolov et al. (2018) [19, Table 2] using the subword CBOW model of Bojanowski et al. (2017) [2] and the 2017 English Wikipedia[15] training corpus (4% of the Common Crawl dataset used by Mikolov et al., 2018) without phrasing. We report English word analogy scores using the $n = 2 \cdot 10^5$ most frequent words, and the case-folding case transformation.

Same as vanilla word vectors The simplest option is to use the initialization of the word vectors $\mathbf{d}_p$ also for the positional vectors $\mathbf{u}_p$: $\mathbf{d}_p \sim \mathbf{u}_p \sim \mathcal{U}(\pm\frac{1}{D})$. In practice, this causes $\mathbf{v}_C \gg \mathbf{w}_C$, decreasing the learning rate (see Figure 1).

Identity positional vectors To ensure $\mathbf{v}_C \sim \mathbf{w}_C$, the simplest option is to initialize the word vectors $\mathbf{d}_p$ to $\mathcal{U}(\pm\frac{1}{D})$ and the positional vectors $\mathbf{u}_p$ to $\mathbf{1}$. Intuitively, the training starts with no positional weighting and positional vectors are learnt later. In practice, $\mathbf{d}_p \ll \mathbf{u}_p$, causing the gradient updates of $\mathbf{d}_p$ to explode for dimensionality $D > 600$. This leads to instability, causing $\mathbf{d}_p = \mathbf{u}_p = \mathrm{NaN}$.

Same as word vectors To ensure $\mathbf{v}_C \sim \mathbf{w}_C$ and $\mathbf{d}_p \sim \mathbf{u}_p$, we require a square distribution $\mathcal{U}^{0.5}(\pm\frac{1}{D})$ such that for i.i.d. $\mathbf{d}_p, \mathbf{u}_p : \mathbf{d}_p \sim \mathbf{u}_p \sim \mathcal{U}^{0.5}(\pm\frac{1}{D})$, we get $\mathbf{d}_p \odot \mathbf{u}_p \sim \mathcal{U}(\pm\frac{1}{D})$. Although an empirical approximate of $\mathcal{U}^{0.5}(0,1)$ using the β-distribution is known [24] (see Figure 2), this does not help with $\mathcal{U}^{0.5}(\pm\frac{1}{D})$, so we need a different approach: If we assume that the context window P is sufficiently large, then $\mathbf{v}_C \sim \mathrm{N}(\mu, \frac{\sigma^2}{|P|})$ by the CLT, where $\mu = \mathrm{E}[\mathcal{U}(\pm\frac{1}{D})] = 0$ and $\sigma^2 = \mathrm{Var}[\mathcal{U}(\pm\frac{1}{D})] = \frac{1}{6D^2}$. For $\mathbf{v}_C \sim \mathbf{w}_C$, we need a distribution X such that $\mathbf{d}_p \sim \mathbf{u}_p \sim X, \mathrm{E}[X^2] = \mu, \mathrm{Var}[X^2] = \sigma^2$. $\mathrm{E}[X^2] = \mu = 0$ leads to $\mathrm{E}[X] = 0$ and $\mathrm{Var}[X^2] = \mathrm{Var}[X]^2$, leading to $\mathrm{Var}[X] = \sigma$. We tested two such X: the uniform $\mathcal{U}\left(\pm\frac{\sqrt[4]{3}}{\sqrt{D}}\right)$ (see Figure 3) and the square-normal $\mathrm{N}^{0.5}\left(0, \frac{1}{\sqrt{6}D}\right)$ [23] (see Figure 4).

[12] https://github.com/tmikolov/word2vec (file word2vec.c, function InitNet)
[13] https://github.com/facebookresearch/fastText (file src/fasttext.cc)
[14] https://github.com/mir-mu/reproducible-ml (file positional-weighting.ipynb)
[15] https://github.com/rare-technologies/gensim-data (release wiki-english-20171001)

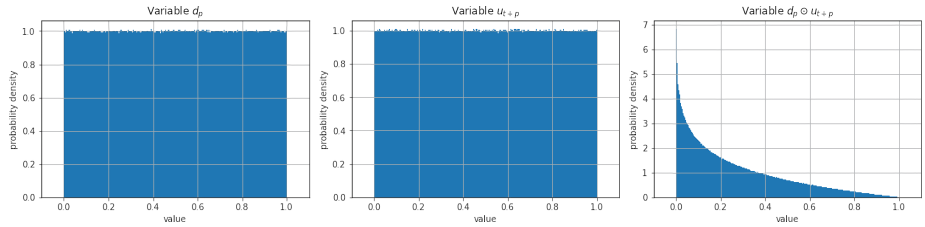


Fig. 1. Probability density functions of values in word vectors $\mathbf{d}_p$ (left), positional vectors $\mathbf{u}_p$ (middle), and their Hadamard products $\mathbf{d}_p \odot \mathbf{u}_p$ (right) with the *same as vanilla word vectors* initialization to $\mathcal{U}(0,1)$. Since $\mathbf{v}_C$ is the average $\mathbf{d}_p$, and $\mathbf{w}_C$ is the average $\mathbf{d}_p \odot \mathbf{u}_p$, we conclude that $\mathbf{v}_C \gg \mathbf{w}_C$, decreasing the learning rate of positional weighting.

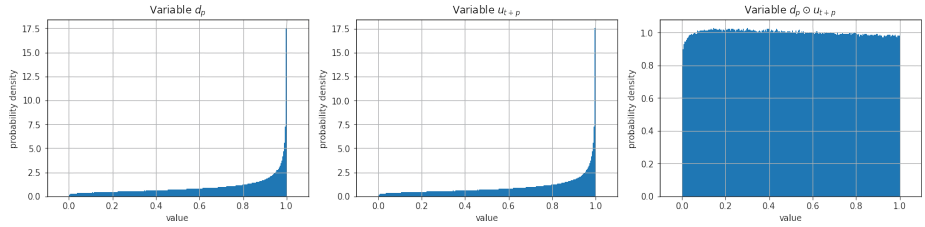


Fig. 2. Probability density functions of values in word vectors $\mathbf{d}_p$ (left), positional vectors $\mathbf{u}_p$ (middle), and their Hadamard products $\mathbf{d}_p \odot \mathbf{u}_p$ (right) with the unused initialization to an empirical approximation of $\mathcal{U}^{0.5}(0,1)$. We need $\mathcal{U}^{0.5}(\pm 1)$ instead of $\mathcal{U}^{0.5}(0,1)$.

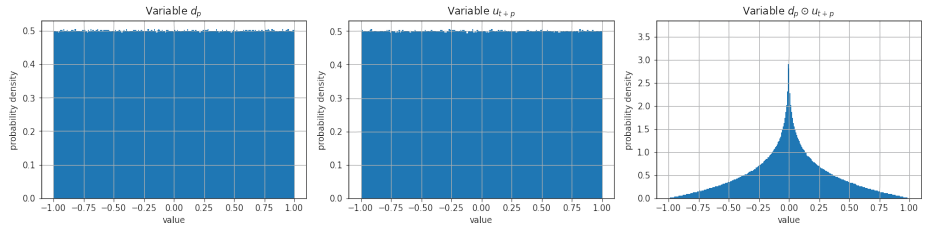


Fig. 3. Probability density functions of values in word vectors $\mathbf{d}_p$ (left), positional vectors $\mathbf{u}_p$ (middle), and their Hadamard products $\mathbf{d}_p \odot \mathbf{u}_p$ (right) with the *same as vanilla word vectors (uniform)* initialization to $\mathcal{U}(\pm 1)$.

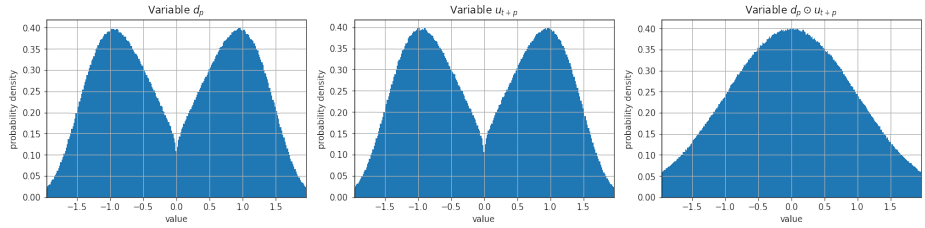


Fig. 4. Probability density functions of values in word vectors $\mathbf{d}_p$ (left), positional vectors $\mathbf{u}_p$ (middle), and their Hadamard products $\mathbf{d}_p \odot \mathbf{u}_p$ (right) with the *same as vanilla word vectors (square-normal)* initialization to a finite-sum approximation of $\mathrm{N}^{0.5}(0,1)$.

Table 3. English word analogy task accuracies and training times of word vectors without positional weighting and with different initializations for positional weighting. The *identity positional vectors* initialization is unstable for dimensionality $D > 600$.

	Accuracy	Training time
No positional weighting	65.52%	2h 06m 33s
No positional weighting (three epochs)	70.94%	4h 41m 17s
Positional weighting, same as vanilla word vectors	50.96%	5h 01m 16s
Positional weighting, identity positional vectors*	75.02%	4h 59m 27s
Positional weighting, same as word vectors (uniform)	74.31%	4h 57m 25s
Positional weighting, same as word vectors (sq.-normal)	74.95%	5h 01m 11s

Table 3 shows that up to 24% of word analogy accuracy can depend on the initialization. The simplest *same as vanilla word vectors* initialization decreases the effective learning rate of positional weighing, leading to a 15% decrease in word analogy accuracy compared to no positional weighting. The second most obvious *identity positional vectors* initialization leads to a 9% increase in word analogy accuracy, but it is numerically unstable for word vector dimensionality $D > 600$. The least obvious *same as word vectors* initializations also achieve a 9% increase in word analogy accuracy, but they are stable for any word vector dimensionality D. Although positional weighting is three times slower, training with no positional weighting for three epochs only leads to a 5% increase in word analogy accuracy, which shows the practical usefulness of positional weighting.

To make results reproducible, we suggest that all papers should report the initialization of weights in their neural networks.

5 Conclusion

With the rapid pace of research in machine learning, the publish-or-perish mantra of academic publishing, and the ever-increasing complexity of language models, maintaining a controlled experimental environment is more difficult than ever. However, identifying and disclosing all confounding variables is important, since it allows us to reproduce and meaningfully compare results.

Our study shows that even simple log-bilinear SVMs contain parameters that are frequently neglected in experiments, although their impact on the results is significant. We believe that more complex machine learning models such as Transformers contain dozens of baked-in parameters and implicit weight initializations that might well be the tipping point towards the singularity.

We hope that our study will make it easier to reproduce both previous and future word vector experiments, and will serve as an inspiration for upholding the principles of reproducibility in future research of machine learning.

Acknowledgments. First author's work was funded by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project.

References

1. Berardi, G., Esuli, A., Marcheggiani, D.: Word Embeddings Go to Italy: A Comparison of Models and Training Datasets. In: Boldi, P., Perego, R., Sebastiani, F. (eds.) Italian Information Retrieval Workshop (IIR 2015). Cagliari, Italy (2015), http://ceur-ws.org/Vol-1404/paper_11.pdf
2. 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)
3. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL pp. 31–40 (2009)
4. Cardellino, C.: Spanish billion word corpus and embeddings (2016), https://crscardellino.github.io/SBWCE/
5. Charlet, D., Damnati, G.: Simbow at SemEval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 315–319 (2017)
6. Chen, X., Liu, Z., Sun, M.: A unified model for word sense representation and disambiguation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1025–1035 (2014)
7. Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H.: Joint learning of character and word embeddings. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
8. Clark, K., Khandelwal, U., Levy, O., Manning, C.D.: What does BERT look at? An analysis of BERT's attention. arXiv preprint arXiv:1906.04341 (2019)
9. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391–407 (1990)
10. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893 (2018)
11. Güngör, O., Yıldız, E.: Linguistic features in turkish word representations. In: 2017 25th Signal Processing and Communications Applications Conference (SIU). pp. 1–4. IEEE (2017)
12. Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., Aluisio, S.: Portuguese word embeddings: Evaluating on word analogies and natural language tasks. arXiv preprint arXiv:1708.06025 (2017)
13. Heaps, H.S.: Information retrieval, computational and theoretical aspects. Academic Press (1978)
14. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International conference on machine learning. pp. 957–966 (2015)
15. Köper, M., Scheible, C., im Walde, S.S.: Multilingual reliability and "semantic" structure of continuous word spaces. In: Proceedings of the 11th international conference on computational semantics. pp. 40–45 (2015)
16. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems. pp. 2177–2185 (2014)
17. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics **3**, 211–225 (2015)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

19. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
20. 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)
21. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26, pp. 2265–2273. Curran Associates, Inc. (2013), https://proceedings.neurips.cc/paper/2013/file/db2b4182156b2f1f817860ac9f409ad7-Paper.pdf
22. Novotný, V., Sojka, P., Štefánik, M., Lupták, D.: Three is better than one. In: CEUR Workshop Proceedings. Thessaloniki, Greece (2020), http://ceur-ws.org/Vol-2696/paper_235.pdf
23. Pinelis, I.: The exp-normal distribution is infinitely divisible. arXiv preprint arXiv:1803.09838 (2018)
24. Ravshan, S.K.: Factor analysis and uniform distributions (2018), https://ravshansk.com/articles/uniform-distribution.html
25. 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), https://is.muni.cz/publication/884893/en
26. Rogers, A., Drozd, A., Li, B.: The (too many) problems of analogical reasoning with word vectors. In: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017). pp. 135–148 (2017)
27. Sen, M.U., Erdogan, H.: Learning word representations for turkish. In: 2014 22nd Signal Processing and Communications Applications Conference (SIU). pp. 1742–1745. IEEE (2014)
28. Svoboda, L., Brychcin, T.: New word analogy corpus for exploring embeddings of czech words. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 103–114. Springer (2016)
29. Unicode Consortium: The Unicode® standard. Mountain view, CA (2020)
30. Venekoski, V., Vankka, J.: Finnish resources for evaluating language model semantics. In: Proceedings of the 21st Nordic Conference on Computational Linguistics. pp. 231–236 (2017)
31. Řehůřek, R.: Subspace tracking for latent semantic analysis. In: European Conference on Information Retrieval. pp. 289–300. Springer (2011)

Part III

Morphology and Syntax

Multilingual Recognition of Temporal Expressions

Michal Starý[1,2], Zuzana Nevěřilová[1,2], and Jakub Valčík[2]

[1] Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech republic
[2] Konica Minolta Laboratory Europe, Holandská 7, Brno, Czech republic

Abstract. The paper presents a multilingual approach to temporal expression recognition (TER) using existing tools and their combination. We observe that the rules based methods perform well on documents using well-formed temporal expressions in a narrower domain (e.g., news), while data driven methods are more stable within less standard language and texts across domains.
With combination of the two approaches, we achieved F1 of 0.73 and 0.9 for strict and relaxed evaluations respectively on one English dataset. Although these results do not achieve the state-of-the-art on English, the same method outperformed the state-of-the-art results in a multilingual setting not only in recall but also in F1. We see this as a strong indication that combining rule based systems with data driven models such as BERT is a valid approach to improve the overall performance in TER, especially for languages other than English.
Further observations indicate that in the domain of office documents, the combined method is able to recognize general temporal expressions as well as domain specific ones (e.g., those used in financial documents).

Keywords: temporal expression, multilingual, date recognition

1 Introduction

Temporal information plays a significant role in NLP tasks such as information retrieval, summarization, question answering or event extraction. The ultimate goal of temporal processing is to extract *events* from unstructured text, i.e., *what* happens, *when* and *how* it relates to some other events.

We divide the ultimate goal into four tasks: 1. temporal expression recognition (TER), 2. temporal expression normalization, 3. event detection, and 4. temporal relation extraction. Attached together these tasks form the temporal processing pipeline that is able to process text to events.

In this work, we focus on *temporal expression recognition* in multilingual cross-domain setting. By *multilingual*, we consider a system able to recognize temporal entities in more than one hundred languages, *cross-domain* means to cover not only generic expressions, but also the temporal expressions specific for specific domains. Even though the practical impact of such universal system

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 67–78, 2020.

is tremendous, the both cross-domain and multilingual setting has received a little attention by a majority of previous work. This is probably caused by the lack of resources with annotated temporal expressions in other languages than English.

We overcome the missing annotated resources by combining two conceptually different approaches. First is the use of a rule based system with multilingual support, HeidelTime. Its multilinguality was achieved by selection and automatic translation of the rules. Second approach is use of BERT NER, a named entity recognition (NER) tool based on multilingual BERT model. Temporal expression recognition is a subset of NER.

We present a combined approach that outperforms the two approaches in recall. We compare the results across different languages and documents types. We point out document types where the combined approach achieves the best results.

2 Related work

Temporal processing was defined in 2002 by specifying the TimeML standard [10]. Further on, temporal processing has been a topic of seven SemEval[3] challenges (2007, 2010, 2013, 2015, 2018, and partly 2016, 2017). As a result, several temporal datasets and temporal processing tools have been developed over last decade.

Later, it was shown that TimeML standard is not sufficient to cover all nuances of temporal information. New annotation standard SCATE was developed [1]. Although being more accurate, it is more complicated. Only two datasets and one tool supporting SCATE annotation schema have been developed so far.

Recent research mostly focuses on (contextualized) word representations and other learning techniques in temporal processing to improve either the performance, universality, or both.

2.1 TimeML and the `TIMEX3` tag

TimeML is a markup language for temporal events in documents. It addresses four problems regarding event markup, including time stamping (that anchors an event to a time), ordering events mutually, reasoning with contextually underspecified temporal expressions, and reasoning about the length of events and their outcomes[4].

In this work, we mostly focus on the time stamping, which is realized through a `TIMEX3` tag. The `TIMEX3` tag[5] is primarily used to mark up explicit

[3] `https://www.aclweb.org/anthology/venues/semeval/`

[4] `http://www.timeml.org`

[5] `http://www.timeml.org/publications/timeMLdocs/timeml_1.2.1.html#timex3`

temporal expressions, such as times, dates, durations, etc. The temporal expressions have values of the following TIMEX3 types: *DATE*, *TIME*, *DURATION*, *SET* (following the TIDES 2002 guidelines [6]).

2.2 Temporal datasets

Several temporal datasets have been developed and made publicly available. They mostly follow the TimeML annotation schema. Concerning the languages covered, English is by far the most covered with datasets from multiple domains. Followed by major Romance languages, German and a few other languages with minor resources. These resources usually contain hundreds to lower thousands of TIMEX3 tags. Not only the number of temporal expressions but also the diversity of them is an important factor of the dataset relevance. This diversity is much higher in cross-domain datasets compared to the single domain ones.

Unfortunately, many of these resources have become unavailable due to lost of support by data providing sites. A presentation of the available temporal datasets used in this work follows.

TBAQ – TimeBank and AQUA [14] is the dataset for temporal processing. It has been developed during TempEval competitions and contains about 200 fully tagged English news documents with 1,243 TIMEX3 tags in total.

TE-3 Platinium [14] is a small dataset containing of only 20 English news documents, that were originally used for evaluating tools developed with access to the TBAQ dataset. It contains 138 TIMEX3 tags.

KRAUTS [13] is a German dataset containing news documents from three different sources, with a high diversity. It contains 1,140 TIMEX3 tags.

PolEval2019 [8] is a recently created Polish dataset that contains more than 1,500 documents from various domains with 6,116 TIMEX3 tags. It is built on top of the Polish KPWr corpus, by taking the named entities annotations and porting those labeled as DATE into TIMEX3 in the TimeML schema.

2.3 Existing Tools

The existing tools for temporal processing are rule based, data driven or hybrid. They also vary in the extent of temporal processing they cover – from focus on TER only to the whole event processing pipeline.

From the rule based category, **HeidelTime** [11] has a growing diversity of supported languages and domains and also ongoing support by creators.

[6] http://www.timeml.org/timeMLdocs/TimeML.xsd

Somewhat similar **SUTime** [3], which is now inferior to HeidelTime, is still used due to its integration in the StanfordCoreNLP package.

Another rule based tool is **SynTime** [17] that uses a set of heuristics over token types to reach state-of-the-art performance on English TimeBank dataset. Moving from rule engineered to data driven tools, **TOMN** [16] is using a conditional random fields that operate on well designed features extracted by predefined pattern matching. A different learning strategy – learning of patterns – have been created by authors of **PTime** [6]. Such a pattern learning was shown to be especially useful in the colloquial domain by setting current state-of-the-art results on English Tweets dataset.

Recent research work used BERT models [5] for TER. On English only, **Chen et al.** [4] have reached comparable results to the existing tools. In the multilingual setting, the fine-tuned BERT model with adversarial alignment by **Lange et al.** [7] has surpassed the previous state-of-the-art multilingual tool HeidelTime with automatic rules by a large margin. Unfortunately, neither of these models has been made publicly available.

3 Multilingual Temporal Expression Recognition

Recent research suggests that monolingual (English) BERT based models perform on English TER comparably to existing tools [4]. Moreover, it has been shown that fine-tuning multilingual BERT model on TER can outperform current state-of-the-art multilingual tools – HeidelTime with automatic rules [12] by a margin. On the languages supported by HeidelTime, the multilingual BERT model was not able to catch up [7]. Also, one has to keep in mind that this multilingual model has been fine-tuned and tested on the same domain (news) and on the same families of languages (Germanic, Romance), therefore the good performance on these languages and domains do not imply good cross-domain and real cross-lingual performance.

We believe that both approaches – rule based and based on BERT have their own advantages in a multilingual setting. The rule based methods work well with well-structured (e.g., *November, 12*) or numeric (e.g., *10/12/2005*) temporal expressions that are easily expressible in terms of rules, either handcrafted or translated. On the other hand, more compound expressions (e.g., *the 2012 through 2016 tax years*, *the end of each year*), expressions with less common modifiers (e.g., *prior year*) or domain specific expressions (e.g., *3rd Quarter*) are handled much better by the multilingual BERT based model.

To exploit them both, we have combined outputs from HeidelTime with outputs from the multilingual BERT based model.

3.1 Rule based HeidelTime [11]

When working in a multilingual setting, the best rule based tool by far is HeidelTime. Not only that it can be seen as a gold standard in the area of TER and normalization, it also has a native support for 13 languages. Event though

there are systems which achieve better scores on English [17,6,16,4], the diversity of supported languages and availability makes HeidelTime still the practical winner.

Moreover, it has been automatically ported by translating the rules to more than 200 languages, and therefore become multilingual in a sense [12]. Nevertheless, the performance of translated rules ranges from bad to extremely bad and is deemed to be useful mostly as a baseline.

Further on, we work with both of these types of rules – the handcrafted as well as the automatically translated.

3.2 Multilingual BERT NER by DeepPavlov

As a representative of BERT based models, we have used publicly available multilingual BERT [5] based NER model by DeepPavlov[7] that was fine-tuned on the OntoNotes dataset [15]. OntoNotes annotations specify 19 tags, from which one is a *DATE* tag with almost 11,000 occurrences in texts from six different domains. Even though the fine-tuning was done solely on English data, the multilinguality stamped into the BERT model during the pre-training [9] allows the model to recognize named entities in 104 languages.

The vast majority of former research was relying on specialized temporal corpora (e.g., the TBAQ corpus). TBAQ contains less then 1,500 annotated temporal expressions from just one domain. Even though it has been shown by Chen et al. that BERT based recognizers already perform well in a low-resource conditions [4], the cross-domain differences are still hard to cover. Higher number of temporal expressions from multiple domains seen during the training shall lead to higher recall, compared to methods that used just TBAQ dataset.

In addition, there are practical advantages of using a generic NER model instead of specific temporal expression tool or custom trained model. The out-of-box functionality (no custom training necessary) and one step extraction of both named entities and temporal expressions are valuable qualities in real world use-cases.

Needless to say, these benefits are not for free, the OntoNotes annotations do not follow the TimeML standard exactly, which means that additional post-processing tasks arise to correctly deal with the type of temporal expressions and with composition and decomposition of recognized expressions.

3.3 The combined approach

To exploit the best of both these approaches we recognize the text by both systems and combine the outputs. We have used a simple algorithm, which prefers recall over precision.

[7] `http://docs.deeppavlov.ai/en/master/features/models/ner.html`

Definition 1. *Let* $pTE = (t, s, e)$ *be a* positional temporal expression *where t stands for the text value, s for the position of the first character of t in the text and e for the last position of t in the text.*

Definition 2. *Let H be a set of positional temporal expressions recognized by HeidelTime and B those recognized by BERT NER.*

To get the final *C* set of combined pTEs we deal distinctly with overlapping and non-overlapping pTEs.

- The combined set C_1 is composed by all pTEs from H that do not overlap with any pTE from B and all pTEs from B that do not overlap with any pTE from H. Simply put, $C_1 = B_{non-overlap} \cup H_{non-overlap}$.
- The combined set C_2 is composed by merging overlapping pTEs together. Merging is done by taking particular overlapping pTEs and creating one longer pTE, that covers all formerly overlapping pTEs.
- Finally, we take the union of both sets to get the final C set, $C = C_1 \cup C_2$

An example result can be seen in Example 1.

Example 1. Bubbly Day

HeidelTime: Pour a glass of sparkling sunshine to celebrate National Bubbly **Day** every first **Saturday** in **June**!

$H = \{(Day, 65, 67), (Saturday, 81, 88), (June, 93, 96)\}$

BERT NER: Pour a glass of sparkling sunshine to celebrate National Bubbly Day **every first Saturday in June**!

$B = \{(every\ first\ Saturday\ in\ June, 69, 96)\}$

$B_{non-overlap} = \emptyset$
$H_{non-overlap} = \{(Day, 65, 67)\}$
$C_1 = \{(Day, 65, 67)\}$
$C_2 = \{(every\ first\ Saturday\ in\ June, 69, 96)\}$

Combined: Pour a glass of sparkling sunshine to celebrate National Bubbly **Day every first Saturday in June**!

$C = \{(Day, 65, 67), (every\ first\ Saturday\ in\ June, 69, 96)\}$

---

4 Evaluation

We divided the evaluation into two categories. For languages supported natively by HeidelTime, we have used the TBAQ-gold and TE3-platinium English and KRAUTS German datasets. For languages not supported natively by HeidelTime, we have used PolEval2019 Polish dataset and to compare with previous work, we have also included German KRAUTS dataset again, but with multilingual setting. We have used evaluation script developed for TempEval-3 [14].

We report the precision, recall and F1 score in both strict and relaxed matching. Strict matching means that both start and end of the entity have to match, whereas relaxed matching requires only partial overlap of predicted entity with the reference one.

For our work, the most important part is the relaxed matching F1 score with additional focus on high recall. These requirements are based on the planned usage of a temporal entity classifier, currently under development. Higher recall leads to recognition of uncommon entities and we can subsequently decide how to deal with them with respect to our goal.

4.1 Monolingual setting

We are using English and German temporal datasets both from the news domain. Note that performance may be different on other domains, since HeidelTime is tailored for the news domain.

Table 1 shows that the BERT NER model performs worse than HeidelTime in both strict and relaxed matching. Also, the performance of the combined model is inferior in strict matching. Still the performance of the combined model in the relaxed matching is comparable to HeidelTime. The combined model achieves the best recall in the relaxed matching.

Table 1. Evaluation in monolingual setting

Language	Dataset	Method	Strict			Relaxed		
			P	R	F1	P	R	F1
English	TBAQ-gold	HeidelTime	**83.64**	**83.32**	**83.48**	**91.68**	91.33	**91.5**
English	TBAQ-gold	BERT NER	72.52	66.9	69.6	90.07	83.1	86.44
English	TBAQ-gold	Combined	69.74	75.91	72.69	86.79	**94.46**	90.46
English	TE3-platinium	HeidelTime	**83.85**	**78.99**	**81.34**	**93.08**	87.68	**90.3**
English	TE3-platinium	BERT NER	76.07	64.49	69.8	92.31	78.26	84.71
English	TE3-platinium	Combined	73.19	73.19	73.19	89.13	**89.13**	89.13
German	KRAUTS	HeidelTime	**80.29**	**65.05**	**71.87**	**90.15**	73.03	80.69
German	KRAUTS	BERT NER	53.92	35.96	43.15	64.94	54.13	64.94
German	KRAUTS	Combined	70.53	64.13	67.18	84.76	**77.06**	**80.73**

4.2 Multilingual setting

We evaluate the approaches in the multilingual setting, since it is an important area for our purposes. We present the result in Table 2.

For the German KRAUTS dataset, we compared the combined method with adversarialy aligned BERT model [7] that was trained on English, Spanish, and Portugese TimeBanks. Even though our combined method outperformed the individual base models, it was inferior to [7].

For the Polish PolEval2019 dataset, we experimented with different HeidelTime settings. Surprisingly, the English HeidelTime outperforms the HeidelTime with automatic German rules by a margin. We believe this is due to a low coverage of the translated rules and a similarity between these languages. Still, the BERT NER outperformed both HeidelTimes. When combined, the HeidelTime with automatic rules turned out to be much better base model. Overall, the combined method reaches the best scores compared to all bases in almost every metric.

We can clearly observe that combining HeidelTime with BERT NER significantly improves the performance compared to the individual models. Still, adversarialy aligned BERT model, fine-tuned on specific temporal datasets from the news domain outperforms our combined method on German news dataset.

Table 2. Evaluation in multilingual setting. N.B. [7] only publishes F1 scores.

Language	Dataset	Method	Strict			Relaxed		
			P	R	F1	P	R	F1
German	KRAUTS	HeidelTime-A	**59.47**	24.5	34.7	**91.31**	37.61	53.28
German	KRAUTS	BERT NER	53.92	35.96	43.15	64.94	54.13	64.94
German	KRAUTS	Combined	57.56	**45.41**	50.77	82.33	**64.95**	72.62
German	KRAUTS	BERT aligned[7]	?	?	**66.53**	?	?	**77.82**
Polish	PolEval2019	HeidelTime (EN)	39.59	19.88	26.47	88.5	44.44	59.17
Polish	PolEval2019	HeidelTime (Auto)	61.13	12.14	20.26	**91.34**	18.14	30.27
Polish	PolEval2019	BERT NER	**62.63**	41.19	49.7	91.2	59.98	72.37
Polish	PolEval2019	Combined (HT-EN)	57.3	42.63	48.89	84.7	63.02	72.27
Polish	PolEval2019	Combined (HT-Auto)	**62.63**	**46.7**	**53.37**	89.89	**67.42**	**77.05**

5 Multilingual TER by document type

We experimented with a collection of English, German and French office documents categorized by their type. For each language, we collected about 1000 documents, divided them into 10 categories with about 100 documents each. These categories are the same throughout the languages.

A preliminary observation has shown that temporal entities in office documents may differ significantly from those contained in other genres such as news. Apart from absolute dates such as *2020/10/31*, office documents contain expressions such as *Q3* meaning *3rd quarter*, or *FY2020* meaning *fiscal year 2020*.

The experiment has two goals:

1. determine what document types contain the most temporal entities
2. which of the three methods is most suitable for TER in these types of documents (again recall is preferred over precision)

We define the *density* of temporal expressions as the number of such entities per 10,000 characters of text. This method is more realistic than the absolute

number since it normalizes absolute numbers per document length. Figure 1 shows the density per document type among English and German documents respectively. It can be seen that for English documents, the category with most temporal entities is project status report and financial report, while for German documents, the highest number of temporal entities is found within invoices and financial reports. It is not surprising the highest number of temporal entities is in financial documents, however, since we have used the combined methods, we need to check that the entities specific to the financial domain were recognized.

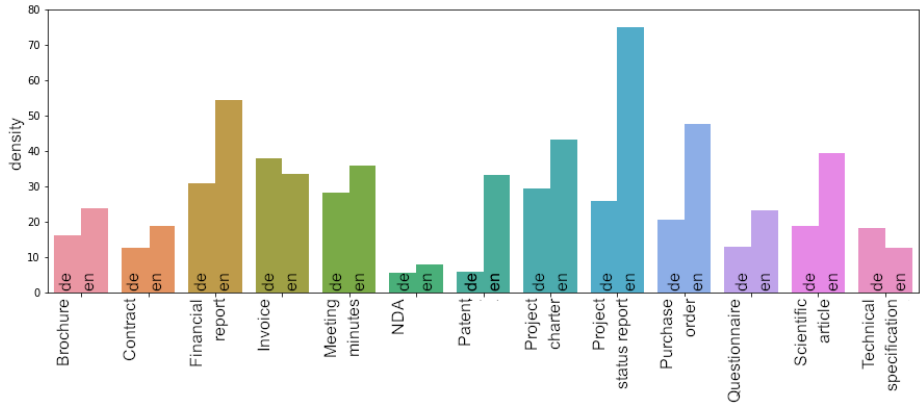


Fig. 1. Number of temporal expressions in English and German documents per document type

We observed that BERT NER is sometimes not able to detect partial dates (e.g., year) when present in an unusual context or, more precisely, in a context typical for office documents. On the other hand, HeidelTime is not able to capture entities from the financial domain since is has not rules for them. The bar charts in Figure 1 shows the number of entities recognized by the combined method.

Examples 2 and 3 show domain-specific temporal entities. The recognized entities are in bold. In English, the combined method is able to recognize such entities. For German, some entities are missed (e.g., *accounting year*) or matched only in some cases (e.g., *financial year*).

Example 2. Temporal entities specific to financial documents

The Company expensed $0.2 million for employee participation in this plan in **fiscal year 2014**.

Lagebericht der AG und Konzernlagebericht für das Geschäftsjahr **2004**.

(Management report of the AG and group management report for the financial year **2004**)

Der strategischen “Wegweiser” wollen wir uns im **Geschäftsjahr 2018** stabil und zukunfts

(We want to be the strategic stable and future “guide” in the **financial year 2018**)

Example 3. Temporal entities specific to legal documents.

For Model numbers 1200, 1300, and 1600 Combo Machines, the warranty is a one **(1) year** parts only, no labor warranty.

Entnehmen Sie dieser Seite alle Informationen zu den **monatlichen** Abschlägen im neuen Abrechnungsjahr.

(On this page you will find all information about the **monthly** payments in the new accounting year.)

6 Conclusion and future work

This paper presents our research of temporal entities recognition in general and within the area of office documents. Since the follow-up task is to extract events related to temporal entities, the algorithm prefers recall over precision.

The main contribution of this work is the combination of rule based and data driven methods. For non-English texts, the combined approach performs the best.

Although we are interested only in some types of office documents (legal and financial), we could not evaluate the performance of the combined methods in this domain because of lack of annotated data.

In further work, we plan to take the following steps.

6.1 Fine-tune BERT NER on temporal datasets

Fine-tuning of BERT model using adversarial alignment as shown by Lange et al. [7] can further improve the performance of our combined system. Still, this fine-tuning has to be done on specialized temporal datasets such as TBAQ. Even though there are ways not to lose the one-step recognition of Named Entities and Temporal Expressions, it remains an open question how will this fine-tuning affect the cross-domain performance.

6.2 Evaluate on more datasets

We believe the differences between rule based and BERT based approach is much more obvious is some domains. Very formal or structured documents shall be well very suitable for rule based HeidelTime. Whereas in domains such as voice assistant or colloquial speech, the BERT based model is deemed to be much better. We aim to quantify these impressions.

6.3 Semantic classification schema

Moving forward from recognition, we are also trying to better understand the temporal expressions. It was shown the TimeML standard is not capturing all important aspects of temporal information [1], so getting the temporal expressions in the TimeML schema is not sufficient. New tagging schemas has been already proposed [1,2], but due to our focus on multilinguality, the tools we are using are following the TimeML or even general NER tagging specifications.

To overcome this issue we are currently working on semantic classification schema, that further granulates the TimeML output of existing systems and, thereby, allows much finer control over detected entities. We see this control as a critical component for improving the performance of a complete temporal processing pipeline.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

1. Bethard, S., Parker, J.: A semantically compositional annotation scheme for time normalization. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 3779–3786. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://www.aclweb.org/anthology/L16-1599
2. Caselli, T., Bartalesi Lenzi, V., Sprugnoli, R., Pianta, E., Prodanof, I.: Annotating events, temporal expressions and relations in Italian: the it-timeml experience for the ita-TimeBank. In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 143–151. Association for Computational Linguistics, Portland, Oregon, USA (Jun 2011), https://www.aclweb.org/anthology/W11-0418
3. Chang, A.X., Manning, C.: SUTime: A library for recognizing and normalizing time expressions. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). pp. 3735–3740. European Language Resources Association (ELRA), Istanbul, Turkey (May 2012), http://www.lrec-conf.org/proceedings/lrec2012/pdf/284_Paper.pdf
4. Chen, S., Wang, G., Karlsson, B.: Exploring Word Representations on Time Expression Recognition. Tech. rep., Microsoft Research Asia (2019)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] (May 2019), http://arxiv.org/abs/1810.04805, arXiv: 1810.04805
6. Ding, W., Gao, G., Shi, L., Qu, Y.: A Pattern-Based Approach to Recognizing Time Expressions. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 6335–6342 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.33016335, http://www.aaai.org/ojs/index.php/AAAI/article/view/4595
7. Lange, L., Iurshina, A., Adel, H., Strötgen, J.: Adversarial Alignment of Multilingual Models for Extracting Temporal Expressions from Text. arXiv:2005.09392 [cs] (May 2020), http://arxiv.org/abs/2005.09392, arXiv: 2005.09392

8. Ogrodniczuk, M., Kobyliński, L. (eds.): Proceedings of the PolEval 2019 Workshop. Institute of Computer Science. Polish Academy of Sciences, Warszawa (2019), oCLC: 1107599814
9. Pires, T., Schlinger, E., Garrette, D.: How Multilingual is Multilingual BERT? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4996–5001. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1493, https://www.aclweb.org/anthology/P19-1493
10. Pustejovsky, J., Castaño, J.M., Ingria, R., Saurí, R., Gaizauskas, R.J., Setzer, A., Katz, G., Radev, D.R.: Timeml: Robust specification of event and temporal expressions in text. In: Maybury, M.T. (ed.) New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA. pp. 28–34. AAAI Press (2003)
11. Strötgen, J., Gertz, M.: Multilingual and cross-domain temporal tagging. Language Resources and Evaluation **47**(2), 269–298 (2013), https://www.jstor.org/stable/42636370, publisher: Springer
12. Strötgen, J., Gertz, M.: A Baseline Temporal Tagger for all Languages. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 541–547. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1063, https://www.aclweb.org/anthology/D15-1063
13. Strötgen, J., Minard, A.L., Lange, L., Speranza, M., Magnini, B.: KRAUTS: A German Temporally Annotated News Corpus. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), https://www.aclweb.org/anthology/L18-1085
14. UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., Pustejovsky, J.: SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). pp. 1–9. Association for Computational Linguistics, Atlanta, Georgia, USA (Jun 2013), https://www.aclweb.org/anthology/S13-2001
15. Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., Houston, A.: OntoNotes Release 5.0 LDC2013T19. Linguistic Data Consortium (October 2013), https://catalog.ldc.upenn.edu/LDC2013T19
16. Zhong, X., Cambria, E., Hussain, A.: Extracting Time Expressions and Named Entities with Constituent-Based Tagging Schemes. Cognitive Computation **12**(4), 844–862 (Jul 2020). https://doi.org/10.1007/s12559-020-09714-8, http://link.springer.com/10.1007/s12559-020-09714-8
17. Zhong, X., Sun, A., Cambria, E.: Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 420–429. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1039, https://www.aclweb.org/anthology/P17-1039

Automatic Detection of Zeugma

Helena Medková

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

Abstract. The term zeugma denotes the linguistic phenomenon that means yoking together expressions with different argument structure, or with semantically incompatible meanings. For that reason, its detection is in many cases relatively difficult subject, mainly because three perspectives of language have to be considered: syntax, semantics and pragmatics. The presented paper describes a possible approach to the detection of this kind of structure. The output of the research will be part of the new online proofreader for Czech.

Keywords: zeugma, grammar checker, VerbaLex.

1 Introduction

Zeugma is the linguistic phenomenon occurring across many languages. Generally, linguists consider it as a stylistic figure in which two or more expressions with different meanings are forced together as in the example (1) [10]:

(1) *She drew a gun and a picture of a gun.*

For that reason, the resulting structure sounds strange or funny. Several linguistic approaches also used so-called zeugma test for the ambiguity recognition. However, this method has proved to be not entirely reliable because an ambiguous expressions do not always lead to zeugmaticity and vice versa [10].

In the Czech language zeugma means in many cases a coordination of elements sharing a common argument. Nevertheless, the dependent expression complies only with one's element argument structure, so the resulting construction is non-grammatical [6].

This paper focuses mainly on non-grammatical sentences where the expressions have different argument structure. An illustrative example is in the sentence (2) where the coordinated elements comprise of predicates. The verb *pocházet* (*come*) should be accompanied with the preposition *z* (*from)* since the only obligatory complement of a verb *pocházet* (*come*) is the genitive prepositional phrase instead of local.

A proper formulation of such structure is a pronominal coreference (2a), where a personal pronoun substitutes the object of the second verb. This strategy also intensifies if the verbs express temporal succession or their meaning is not analogous [4].

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 79–86, 2020.

(2) *Pocházím a bydlím v Praze.* (*I come and live in Prague*)

(2a) *Pocházím* (*od / z*) *Prahy a žiju v ní.* (*I come from Prague and live in it.*)

When processing this task, it is necessary to distinguish semantics that sets verbal argumentative structure. Let us take the verb *navazovat*[1] as the example. In meaning *to follow up* something, it is essential to use the prepositional phrase. Otherwise, if it is used in a sense *to establish something*, it requires the object in the accusative (cf. sentences (3), (4)).

(3) *Navazuje a rozšiřuje publikaci…* (*Follows up and extends publications…*)

(4) *Mohou navazovat a prohlubovat své kontakty mezi sebou …* (*They can establish and deepen their contacts with each other…*)

Zeugma recognition is challenging for verbs with an implicit (zero) object because of utterance ambiguity. The knowledge of the author's intentions in the utterance context is fundamental for its correct interpretation. In these cases, it is complicated to determine, in particular, whether the coordinated verbs have a common argument.

While in the example (5) it is possible to find out quite clearly from the context of the verb *číst* (*read*) that the verbs share the same object, in the example (6) it is not accessible to judge. Without further knowledge of the plotted situation, it is not possible to determine precisely whether the pupils read the same text with which they later worked in the lesson.

(5) *… byť Chrome umí číst a pracovat s lokálními médii…* (*… although Chrome can read and work with local media…*)

(6) *Žáci v předchozí hodině četli a pracovali s textem na téma životní prostředí – třídění odpadů.* (*Pupils in the last lesson read and worked with the text about the environment – waste sorting.*)

2 Rule-based Detection of Zeugma

This paper presents a rule-based approach to zeugma detection. It involves creating a grammar with specific rules that check if there is a proper object in a proper form for a particular verb in the context of the sentence.

If the rule does not find a grammatical addition, parser marks structure as zeugma and send it to output.

The rules are focused mainly on the verb that has an incorrect phrase in a given sentence. In most cases, rules detect the unsuitable dependent located at the postposition of the coordination (as in the examples (2), (3), (5)). It implies that it is mainly the first verb in the conjunction, which binding needs to be corrected.

[1] Two meanings: *to follow up*, *to establish something*

The first rules were manually created within the diploma thesis [1] for 83 verbs. The disadvantage of this approach was that the rules covered only the non-grammatical structures for the verbs specified in the variables.

However, in the Czech lexicon, there are thousands of verbs. E.g. the lexical database VerbaLex [2] contains 10469 verbal lemmas and 19247 verbal patterns. It would be tedious to manually create rules for all the verbs contained in Czech lexicon.

Therefore, new grammar was generated based on manual grammar according to the verbal patterns in VerbaLex [2].

2.1 VerbaLex Processing

The intermediate step to obtain automatically generated grammar in this approach is to create a better processable data structure from the database. At first, it is necessary to get all single verbs from VerbaLex synonymous series

dok[2]: ověřit/ověřit si[3]:1 ned: ověřovat/ověřovat si:1

and its relevant parts of the frames (with an argument structure after VERB in right periphery)

+AG(kdo1[4];<person:1>;obl)+++VERB+++INFO(co4[5];<fact:1>;obl)

from the database into a dictionary):

```
{'ověřit': {'transitivity': '=canbepassive: yes',
            'first_objects_frames':
            [['INFO(co4;<fact:1>;obl)\n'],
            ['INFO(že;<info:1>;obl)'], [...]]}}
```

Grammar is generated after this processing phase.

2.2 Generated Grammar Organization

The grammar contains rules for four possible sentence arrangements, where zeugma can be recognized. Each of the four arrangements aggregates verbs with the same obligatory complements at the position of the first subject. There are 649 such aggregates in grammar, containing together 5300 verbs. There is also a potential for further expansion when we resolve the component issues associated with the processing.

---

[2] Aspect: pf. / impf.
[3] to verify
[4] nom_anim
[5] acc_inanim

An example of the generated rule. Rule illustration represents one aggregate with four potential structures that can be recognized.

In the variable *$verb(lemma)* are merged all verbs with the first obligatory accusative object in valency pattern according to VerbaLex.

The transitive verbs (in lexical database have label *=canbepassive: yes*, require another condition (*$verb(tag not): k5.*mN.**) to pass out all passive forms. Tags *bound* and *rbound* signal segment boundaries.

```
TMPL: bound $context* $verb (word a) (tag k5.*) $prep $noun
$context* rbound AGREE 2 4 mgn MARK 2 4 5 6 <zeugma>

TMPL: bound $context* $verb (word a) (tag k5.*) $refl $prep
$noun $context* rbound AGREE 2 4 mgn MARK 2 4 5 6 7 <zeugma>

TMPL: bound $context* $verb (word a) (tag k5.*) $refl $noun
$context* rbound AGREE 2 4 mgn MARK 2 4 5 6 <zeugma>

TMPL: bound $context* $verb (word a) (tag k5.*) $noun
$context* rbound AGREE 2 4 mgn MARK 2 4 5 <zeugma>

$verb(lemma): rdousit pohněvat setřepat ověřit ověřovat ...
$verb(tag not): k5.*mN.*
$prep(tag): k7.*
$context*(tag not): k3.*yF.* .*c4.*
$noun(tag not): k3.*yF.* .*c4.*
$noun(tag): k[123].*
```

The method is currently unreliable when the null object (6) occurs, and also in cases of the elided object as seen on (7). The antecedent of the verb *odmítnout* (*refuse*) is expressed in the first clause of the compound sentence. Therefore the rule recognizes it as zeugma.

(7) *Po bulharské okupaci Makedonie dostal pozvání do probulharské vlády, ale odmítl a zapojil se do komunistického odboje.* (*After the Bulgarian occupation he got an invitation to probulharic government, but he refused and joined to communistic resistance.*)

Nevertheless, there is a possibility to recognize whether two verbs in a sentence have a common object. One of the solutions involves statistics using collocations. As can be seen in the example (7), the verb *odmítnout* (*refuse*) is not semantically compatible with the complement *do komunistického odboje* (*into the communist resistance*) of the verb *zapojit* (*join*).

With the knowledge of the verb and object collocability, it will be possible to determine whether the object belongs to both predicates in coordination or only to one. It will help to get more accurate results.

An alternative solution to this is the creation of a text preprocessing tool that can learn how to classify coordinations, where the dominant expression shares a common addition.

The tools of Nature language processing centre (CZPJ FI MU) are applied for detection purposes. Namely the morphological tagger Majka [9], disambiguator Desamb [8] and SET parser [7] are used for syntactical analyses.

3 Standard Data Set for Zeugma Detection

The data set consists of the sentences manually selected from the cztenten17 [5] corpus. As the zeugma is relatively challenging to find, an error was intentionally made in 17 sentences. Although there was an effort to get all of the sentences authentic, two clauses were utterly made up. Labels *korpus*, *upraveno*, * (*corpus, edited*, *) indicate the origin of the sentences.

The basis of data set forms 750 sentences with zeugma (83 various verbs) collected within the diploma thesis *Automatic detection of non-grammatical constructions in Czech* [1].

The original data set was annotated and expanded to 1013 positive cases of zeugma. To each verb in incorrectly coordinated construction were also added ca. 20 negative ones (giving a total number of 1681). 1137 of those sentences containing coordination or zeugma also have concurrent ones, which provide additional context to them. The whole data set now comprises of 5313 sentences.

Table 1. Data set – statistical data

Data set statistics	Count
Sentences in data set	**5313**
Sentences with correct coordination	**1681**
Sentences with Zeugma	**1013**
Words	**79379**
Verbs	**84**

In the data set, there are only zeugmas formed by verbs connected by the conjunction 'a'. Therefore there is a necessity to enlarge and balance it in the future. Nevertheless, the data set makes possible monitoring the quality of rules even now.

3.1 Evaluation of Grammars

The python script automatically evaluates grammar quality. As the input files, it takes .csv data set and the SET output file. The SET parser output consists of the defective coordinations labelled as <*zeugma*> and the whole sentence with label <*sentence*>:

<zeugma> (k5eAaImIp3nS): zkoumá a bádá nad artefakty
<sentence> (kIx.): Archeolog zkoumá a bádá nad artefakty .[6]

The program converts the SET output into the matrix with actual values (1 – positive and 0 – negative), as it also similarly evaluates the data set with predicted values. The script then creates the confusion matrix and counts the required measures.

Table 2. Data set – comparison of evaluations

Grammar	Precision	Recall	F-score	Accuracy
Manual grammar	**0.982**	**0.381**	**0.549**	**0.765**
Generated grammar	**0.796**	**0.224**	**0.350**	**0.687**

Grammar control requires the most accurate results in the first place. That is the reason why the precision measure is for proofreader [3] purposes more important than the recall. As seen in table 2, manual grammar has on the data set relatively accurate results.

However, testing on the part of czTenTen17 corpora in diploma theses [1] showed precision score 0,633 (without mistakes in morphological tags, it was 0,869). Enlarging data set primarily with negative cases of zeugma will provide significantly more reliable results.

Although automatically generated rules in the grammar are generic, without special conditions for achieving better scores, testing on the data set has shown promising results for further experiments with this approach.

However, there is a number of issues for consideration. The first significant problem is recall. Manually created grammar covers approximately 38 % of defective structures. The aim was to get more precise results for the price of lower coverage. Generically created rules have even ten percent lower recall. This approach allows covering a more considerable amount of Czech verbs. However, the set restrictions in the rules do not enable matching each structure in the data set. It also significantly reduces the recall.

Grammar does not allow finding structures as in example (8). The current rule is limited with the condition, that there must be no suitable addition for the verb *doporučit* (*to recommend*) in the whole sentence context.

(8) *Doporučíme a vybereme s vámi rostliny...* (*We recommend and choose with you the plants...*

(9) *... zde bych doporučil a odkázal na Castanedu* (*... in this place I would recommend and refer to Castaneda*

[6] An archaeologist examines and researches artifacts.

Also in example (9) is shown, that rules do not recognize the zeugma, when the expected object of the first verb is in the accusative, but the predicates share a prepositional phrase with the dominant noun in the accusative (or vice versa (10)).

(10) *Dohlíží a prověřuje faktury....* (*He supervises and checks out the invoices...*)

In the future, it is necessary to focus on enlargement of the rules patterns and also choosing contextual restrictions in a more targeted way.

Testing on the czTenTen17 corpus revealed a large amount of errors in morphological tags. Contextual ellipses or vaguely formulated rules caused another relevant errors.

4 Summary and Future work

This paper has presented the rule-based approach to zeugma detection that proposes a manual grammar, and also grammar automatically generated from the lexicon of verb valencies VerbaLex. The standard annotated data set was created for purposes of this research, that extended earlier data from the master thesis. Further work involves mainly enlargement of the current data set. The inclusion of semantic perception, with the usage of semantic roles in VerbaLex, collocations or machine learning, also has to be considered. The increment of recall and precision of grammar is necessary as well.

Acknowledgements. This work was supported by the project of specific research *Čeština v jednotě synchronie a diachronie* (Czech language in the unity of synchrony and diachrony; project no. MUNI/A/0913/2019) and by the Technology Agency of the Czech Republic under the project TL02000146.

References

1. Geržová, H.: Automatická detekce negramatických větných konstrukcí pro češtinu (in Czech, Automatic detection of non-grammatical constructions in Czech). Master's thesis, Masaryk University, Faculty of Arts, Brno (2019 [cit 2020-10-30]), `https://is.muni.cz/th/fuz2y/`
2. 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). ISBN 80-224-0895-6.
3. Hlaváčková, D., Hrabalová, B., Machura, J., Masopustová, M., Mrkývka, V., Valíčková, M., Žižková, H.: New Online Proofreader for Czech. In: Horák, Aleš; Rychlý, Pavel; Rambousek, Adam (eds.): Slavonic Natural Language Processing in the 21st Century. p. 79-92, 14 pp. Brno: Tribun EU (2019). ISBN 978-80-263-1545-2.
4. Hrbáček, J.: Společný předmět u dvou slovesných přísudků (in Czech, Two verbal predicates with common object). Naše řeč. **47**(2), p. 118–120. Ústav pro jazyk český AV ČR, (1964). `http://nase-rec.ujc.cas.cz/archiv.php?lang=en&art=5022`

5. 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, July)
6. Karlík, P.: Zeugma. In: Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), CzechEncy – Nový encyklopedický slovník češtiny, (2017). https://www.czechency.org/slovnik/ZEUGMA Last accessed 26 Oct 2020
7. 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. p. 161-171, 11, pp. Berlin/Heidelberg: Springer (2011). ISBN 978-3-642-20094-6.
8. Šmerk, P.: K morfologické desambiguaci češtiny (in Czech, Towards morphological disambiguation of Czech). PhD thesis, Masaryk University, Faculty of Informatics, Brno (2007)
9. Šmerk, P.: K počítačové morfologické analýze češtiny (in Czech, Towards Computational Morphological Analysis of Czech). Ph.D. thesis, Faculty of Informatics, Masaryk University (2010)
10. Viebahn, E.: Ambiguity and Zeugma. Pacific Philosophical Quarterly. 99. 10.1111/papq.12229, (2018). https://onlinelibrary.wiley.com/doi/pdf/10.1111/papq.12229?casa_token=

Cthulhu Hails from Wales

N-gram Frequency Analysis of R'lyehian

Vít Novotný and Marie Stará

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{413827,409729}@mail.muni.cz

A'bstract R'lyehian is a unique fictional language penned by the prolific 20th century horror fiction author H. P. Lovecraft. Prior work in the area of the Lovecraftian mythos has not yet studied the similarities between R'lyehian and natural languages, which are crucial for determining its true origins. We produced a comprehensive wordlist of R'lyehian and used open-source *N*-gram-based language identification tools to find the most similar natural languages to R'lyehian. From the comprehensive wordlist, we also constructed a frequency table of all unigraphs and digraphs in R'lyehian. We show that R'lyehian is most similar to Celtic languages, which lays grounds for our hypothesis that R'lyeh, where Cthulhu lies dreaming, might be a place in Wales. Our frequency tables will prove a useful resource for future work in the area of the Lovecraftian mythos.

K'eywords: H. P. Lovecraft, language identification, *N*-grams, R'lyehian

1 I'ntroduction

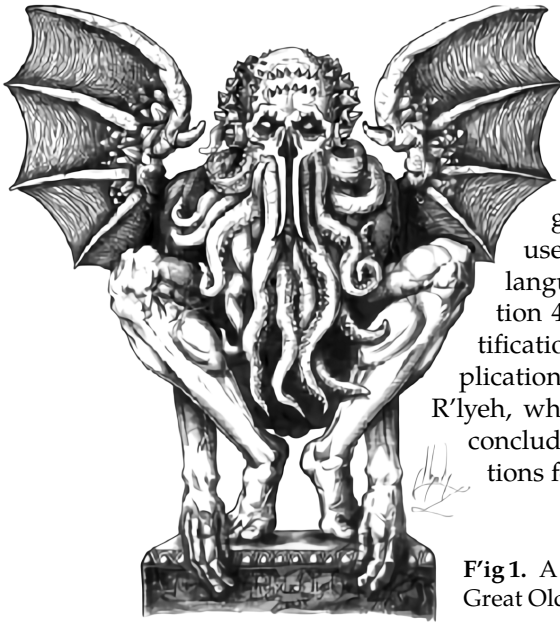
H. P. Lovecraft is regarded as one of the most influential authors of the 20th-century horror genre. R'lyehian is a fictional language spoken by ancient cosmic dieties (the *Great Old Ones*, see F'igure 1) in Lovecraft's 1926 short story *Call of Cthulhu* [7] and in his later work. Below is an example sentence in R'lyehian:

> *Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn.*
> In his house at R'lyeh dead Cthulhu lies dreaming.

Prior work in the area of the Lovecraftian mythos has neglected the similarities of R'lyehian to natural languages and has focused mainly on Lovecraft's use of English. [5,13] Since R'lyehian has been romanized, it lends itself to character *N*-gram frequency analysis and therefore also language identification.

Prior work has not determined the exact location of the sunken city of R'lyeh. Lovecraft (1928) [7] places R'lyeh at 47°9'S 126°43'W in the southern Pacific Ocean, whereas Derleth (1952) [4], a correspondent of Lovecraft, places R'lyeh at 49°51'S 128°34'W. By identifying the most similar natural languages to R'lyehian, we hope to discover the true location of the resting place of the Old One Cthulhu.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 87–92, 2020.



Our work is structured as follows: In Section 2, we briefly discuss R'lyehian and its phonology. In Section 3, we describe our romanized wordlist and the open-source N-gram-based language identification tools that we used to find the most similar natural languages to written R'lyehian. In Section 4, we present the language identification results and discuss their implications for locating the sunken city of R'lyeh, where Cthulhu lies dreaming. We conclude in Section 5 and suggest directions for our future work.

F'ig 1. A monolith of Cthulhu, one of the Great Old Ones (Alexander Liptak, CC BY). [3]

2 R'lyehian

R'lyehian, also known as Cthuvian, is a language created by H. P. Lovecraft for his 1926 short story *Call of Cthulhu* [7]. Unlike some other fictional languages, such as the J. R. R. Tolkien's Elvish languages,[1] or Marc Okrand's Klingon [10] from the Star Trek universum, Lovecraft's R'lyehian appears only in fragments and has no comprehensive vocabulary or grammar.

Below, we list a few f'acts known about R'lyehian:

- It is (supposed to be) unpronounceable for humans. [12]
- As it uses a number of different prefixes and suffixes, it can be classified as a synthetic language.[2] Unfortunately, not enough data exist to subclassify it more accurately as either an agglutinative or a fusional (inflected) language.
- It makes no distinction between the past and the future, only the present and the non-present, [9] since the Old Ones exist at all times simultaneously.
- It does not distinguish parts of speech and has free word order. [11,1]
- It is written in hieroglyphics, see e.g. the text on the pedestal in F'igure 1. [9] The romanized spelling reflects how English speakers captured the speech.[3]

[1] Tolkien's Sindarin was partially based on Welsh, which Tolkien discusses in his 1931 essay *A Secret Vice* [14].

[2] https://en.wikipedia.org/wiki/Synthetic_language

[3] "[The word Cthulhu is] supposed to represent a fumbling human attempt to catch the phonetics of an *absolutely non-human* word." (Lovecraft, 1976) [8, pp. 10–11]

Some useful insights about R'lyehian can be found in the work of Robinson (2010) [12] describing the names (teratonyms[4]) used by H. P. Lovecraft. Robinson describes the features Lovecraft used to make the language seem unpleasant and harsh as well as the influence of other languages (Arabic, Hebrew, and fragments of African languages) on teratonyms. Some of their conclusions can be applied to the language of R'lyehian as a whole.

The intentional strangeness of R'lyehian language was, according to Robinson, produced at three levels:

1. individual sounds,
2. sound combinations, and
3. word-forms.

At the first level, the strangeness was produced by clustering consonants atypical for English, such as the aspiranted consonants or various nasal combinations, e.g. *bn*, *mn*, *mt*, *mth*, or *pn*.

At the second level, the unpronouncibility was produced similarly to the first level by creating clusters unnatural for English or by using clusters that appear in English, but placing them "in patterns or positions that run contrary to its phonotactics". For example: beginning a syllable with a cluster that usually appears at the end of English words, such as *pth* in *depth*.

As for the third level, it can be stated simply by looking at the words in R'lyehian that it seems and sounds unnatural and strange. To achieve this goal, Lovecraft used low (*a*) and back (*u*, *o*) vowels and consonants that are perceived as harsh and dissonant.

2.1 P'ronounciation

There are no clear rules for pronouncing R'lyehian. To the best of our knowledge, Lovecraft himself described merely the pronounciation of the name *Cthulhu*:

> "The actual sound — as nearly as human organs could imitate it or human letters record it — may be taken as something like *Khlûl'hloo*, with the first syllable pronounced gutturally and very thickly. The *u* is about like that in *full*; and the first syllable is not unlike *klul* in sound, since the *h* represents the guttural thickness. The second syllable is not very well rendered — the *l* being unrepresented." (Lovecraft, 1976) [8, p. 11]

3 M'ethods

To identify the most similar natural languages, we required a corpus or a wordlist of R'lyehian and an *N*-gram-based language identification tool with pre-trained models for natural languages. In this section, we present our comprehensive wordlist and our frequency table of all unigraphs and digraphs in R'lyehian, and the language identification tools that we used in our experiment.

[4] Names of monsters: *terato* (monster) + *nym* (name)

3.1 R'lyehian wordlist

Due to the sparse occurences of R'lyehian in Lovecraft's work, we decided against producing a R'lyehian corpus. Instead, we collated two online resources [11,1] into a comprehensive wordlist that we show below in alphabetical order:

1. *ah*
2. *athg*
3. *bug*
4. *bugg-shoggog*
5. *cf'ayak*
6. *cf'tagn*
7. *chtenff*
8. *cthugha*
9. *cthulhu*
10. *ebumna*
11. *ee*
12. *ehye*
13. *ep*
14. *farnomi*
15. *fhtagn*
16. *fhthagn-ngah*
17. *fm'latgh*
18. *fomalhaut*
19. *ftaghu*
20. *geb*
21. *gnaiih*
22. *gof'nn*
23. *goka*
24. *gotha*
25. *grah'n*
26. *h'ehye*
27. *hafh'drn*
28. *hai*
29. *hastur*
30. *hlirgh*
31. *hrii*
32. *hupadgh*
33. *iä*
34. *ilyaa*
35. *k'yarnak*
36. *kadishtu*
37. *kn'a*
38. *li'hee*
39. *llll*
40. *lloig*
41. *lw'nafh*
42. *mg*
43. *mglw'nafh*
44. *mnahn'*
45. *n'gai*
46. *n'gha'ghaa*
47. *n'gha-ghaa*
48. *n'grkdl'lh*
49. *n'gha*
50. *n'ghft*
51. *naf'lthagn*
52. *nglui*
53. *nilgh'ri*
54. *nog*
55. *nw*
56. *ooboshu*
57. *orr'e*
58. *ph'nglui*
59. *ph'nglui*
60. *phlegeth*
61. *r'luh*
62. *r'lyeh*
63. *ron*
64. *s'uhn*
65. *sgn'wahl*
66. *shagg*
67. *shogg*
68. *shtunggli*
69. *shugg*
70. *sll'ha*
71. *stell'bsna*
72. *syha'h*
73. *tharanak*
74. *thflthkh'ngha*
75. *throd*
76. *uaaah*
77. *uh'e*
78. *uln*
79. *ulnagr*
80. *vugtlag'n*
81. *vugtlagln*
82. *vulgtlagln*
83. *vulgtm*
84. *vulgtmm*
85. *wgah'n*
86. *wgah'nagl*
87. *y'bthnk*
88. *y'hah*
89. *ya*
90. *ygnailh*
91. *yog-sothoth*
92. *yuggoth*
93. *zhro*

From the wordlist, we extracted the affixes of R'lyehian:

1. *-agl*
2. *-agn*
3. *-agr*
4. *-nyth*
5. *-og*
6. *-or*
7. *-oth*
8. *-yar*
9. *c-*
10. *h'-*
11. *na-*
12. *nafl-*
13. *ng-*
14. *nnn-*
15. *ph'-*
16. *y-*

From the wordlist, we also constructed a frequency table of all unigraphs and digraphs in R'lyehian in T'able 1. Our table shows that R'lyehian consists of 7 vowels and 28 consonants, including 11 digraphs mostly created by the consonant $+h$, which changes the pronunciation of the first consonant.

3.2 L'anguage identification

In this section, we describe the open-source language identification tools that we used in our experiment. Our selection is based on the survey of Jauhiainen et al. (2019) [6]. We report the top three languages identified by the tools.

T'able 1. Frequencies of all unigraphs and digraphs in R'lyehian extracted from our comprehensive wordlist. We categorize the unigraphs into consonants and vowels.

Unigraphs				Digraphs	
Consonants		Vowels			
g	9.06%	*a*	12.33%	*th*	2.89%
n	7.90%	*'*	7.71%	*gh*	2.31%
l	7.51%	*u*	5.59%	*ng*	1.35%
h	5.39%	*o*	4.05%	*sh*	1.35%
r	3.47%	*i*	3.85%	*fh*	0.96%
t	3.08%	*e*	3.47%	*lh*	0.77%
f	2.31%	*ä*	0.19%	*ph*	0.58%
y	2.31%			*ch*	0.19%
m	1.93%			*kh*	0.19%
k	1.73%			*yh*	0.19%
s	1.54%			*zh*	0.19%
b	1.35%				
w	1.16%				
d	0.96%				
v	0.96%				
c	0.77%				
p	0.39%				

T'able 2. The top three closest natural languages to R'lyehian identified by three different language identification tools. Celtic languages are *emphasized*.

Tools	Languages
TextCat	*Scots, Manx, Welsh*
CLD2	*Irish*, Croatian, Sesotho
LangDetect	Somali, Indonesian, *Welsh*

TextCat In their seminal work, Cavnar et al. (1994) [2] described the *out-of-place* N-gram-based language identification method, which is implemented by the open-source TextCat tool.[5] TextCat contains models for 69 natural languages.

CLD2 Compact Language Detector 2[6] (CLD2) is the language identifier from the Google Chrome web browser. For Unicode blocks that map one-to-one to detected languages, CLD2 uses simple rules. For others, CLD2 uses a Naive Bayes classifier on character N-grams. CLD2 contains models for 160 natural languages.

LangDetect LangDetect[7] is a language identifier that uses a Naive Bayes classifier on character N-grams. Like CLD2, LangDetect applies a number of normalization heuristics to the input text. LangDetect supports 55 natural languages.

4 R'esults

T'able 2 places R'lyehian closest to Celtic languages (Scots, Manx, Welsh, and Irish) with Welsh being the most frequent among the top three closest languages. As a result, we hypothesize that R'lyeh might be the Caldey Island in Wales at 51°38'N 4°41'W, where hooded monks observe Celtic rites and make offerings of the darkest of chocolates to the slumbering Cthulhu.

[5] https://www.let.rug.nl/~vannoord/TextCat/
[6] https://pypi.org/project/cld2-cffi/
[7] https://pypi.org/project/langdetect/

5 C'onclusion

Although Lovecraft's fictional language of R'lyehian is purposely distinct from natural languages, our results suggest that R'lyehian was inspired, either consciously or subconsciously, by the Celtic language of Welsh.

Future work should compare the phonology of Welsh and R'lyehian using our comprehensive wordlist and frequency table of all unigraphs and digraphs, and expand our wordlist by manning a Wales expedition to interview Cthulhu.

A'cknowledgments. First author's work was funded by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project.

R'eferences

1. Admin of Naguide.com: Call of Cthulhu – R'lyehian Language Guide (November 2018), https://www.naguide.com/call-of-cthulhu-rlyehian-language-guide/
2. Cavnar, W.B., Trenkle, J.M., et al.: *N*-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval. vol. 161175 (1994)
3. Cole, D.R., et al.: Cthulhuic Literacy: Teaching Secondary English with a Dose of Lovecraft. English in Australia **49**(1), 72 (2014)
4. Derleth, A.: The Black Island. Weird Tales (January 1952)
5. Jamneck, L.: Tekeli-li! Disturbing Language in Edgar Allan Poe and H. P. Lovecraft. Lovecraft Annual (6), 126–151 (2012), https://www.jstor.org/stable/26868454
6. Jauhiainen, T.S., Lui, M., Zampieri, M., Baldwin, T., Lindén, K.: Automatic language identification in texts: A survey. JAIR **65**, 675–782 (2019)
7. Lovecraft, H.P.: The Call of Cthulhu. Weird Tales (February 1928)
8. Lovecraft, H.P.: Selected Letters [V] 1934–37. Arkham House (1976)
9. Luethke, K.: Fathoming the Unknown: A Divulge into H. P. Lovecraft's Use of Linguistic Phonology and Entomology in Relation to Cosmic Horror and the Cthulhu Mythos. The Luethke Company (2014)
10. Okrand, M.: The Klingon Dictionary: The Official Guide to Klingon Words and Phrases. Simon and Schuster (1992)
11. Roadagain, R., Haq, A., et al.: R'lyehian (November 2020), https://lovecraft.fandom.com/wiki/R'lyehian
12. Robinson, C.L.: Teratonymy: The Weird and Monstrous Names of HP Lovecraft. Names **58**(3), 127–138 (2010), https://doi.org/10.1179/002777310X12759861710420
13. Spencer, H.: Semantic Prosody in Literary Analysis: A Corpus-based Stylistic Study of H. P. Lovecraft's stories. Master's thesis, University of Huddersfield (2011)
14. Tolkien, J.R.R.: The Monsters and the Critics, and Other Essays. George Allen and Unwin (1986)

Part IV

Text Corpora

RapCor, Francophone Rap Songs Text Corpus

Alena Podhorná-Polická

Faculty of Arts, Masaryk University
Arna Nováka 1, 602 00 Brno, Czechia
podhorna@phil.muni.cz

Abstract. The paper introduces the RapCor corpus, which is a specific text corpus for French, based on francophone rap songs' texts from the last three decades when rap music became one of most popular music genres. An overview of more than ten years of rap corpora building presents our motivations, text processing methods, annotation decisions, as well as achievements and problematic issues. The published part of rap corpora, available in Sketch Engine manager for interdisciplinary research, the RapCor 1288, consists of 709,057 words of 1288 francophone rappers' texts. It had been used mainly for the detection and longitudinal observation of so-called "identitary neologisms", i.e. expressions emerging from communication between peers, motivated by search for group belonging, playfulness and expressivity. Rappers' language is also a valuable resource for investigating metaphors and idioms that have been formed by assigning a new meaning to existing language items. The main goal of this largely substandard linguistic corpora is to uncover the phonemic and semantic innovations and trends in modern French.

Keywords: French corpus; text processing; rap music; hip hop; lyrics; substandard language; neology; written orality; corpus building

1 Introduction

Text corpora are useful resources for investigation into a broad range of linguistic issues, including the study of neologisms in a short-term diachronic perspective. This approach became more accessible recently thanks to the arrival of big data corpora (e.g. [12]). Together with gradual perfection of neologism detection tools based on typical co-occurrence analysis (so called "discriminants", see [8]), or, more usually, based on exclusion lists or dictionaries (e.g. European project *Néoveille*, cf. [1]), the automatic extraction of neologism candidates seems to be able to facilitate the detection of ongoing changes in the lexicon (for an overview in Czech, see [17]).

However, our previous collaboration with the Lingea publishing house on the second edition of the substandard French-Czech dictionary *Pas de blème !* [15] and our ongoing participation in Neoveille's project confirms our conviction that there is still a lot to do as regards the detection of so-called "identitary neologisms", i.e. expressions emerging from communication between peers, motivated by search for group belonging, playfulness and expressivity [11]. It

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 95–102, 2020.

is due to: a) its low frequency in newspaper monitor corpora which are usually used to neo-crawling, b) difficulties in automatic detection of neosemantisms (changes of meaning of existing word forms, as is very frequent in the formation of new words in slang) and c) gaps in integrated exclusion dictionaries that are rarely based on spoken language vocabularies. Thus, in order to fill these gaps and being motivated by pedagogical and translation considerations, we set out to compile a specific slang corpus for French with a focus on rap music lyrics, well known for a high frequency of slang words [4, 20], as well as a variety of identitary neologisms in early stages of their societal diffusion.

2 Corpus Construction Motivations

More than ten years ago, the idea of a rap corpus emerged in the Department of Romance Languages and Literatures of Masaryk University, motivated by the then research goal to track identitary neologisms (e.g. *bolos* [10]) as well as by the students' constant interest[1] in discovering substandard French via rap lyrics, mainly through translation or linguistic analysis in their scholar theses.

At that time, French rap was growing exponentially until it became the second largest market for the production and consumption of this genre after the American scene [18]. Nowadays, rap music, as one of hip-hop culture pillars (together with break dance, DJing, beat-box and graffittis), dominates French music industry despite several controversies that have even led to lawsuits [5].

Over the past four decades, rappers have been reflecting historical, political, and economic circumstances, focusing the listeners' attention to their own origins, usually but not exclusively those emblematic districts and suburbs (called *cités* in French), which are ethnically mixed and geographically more or less segregated. But rap lyrics also describe everyday life, personal problems, dreams and visions along with other features, characteristic for youth culture, such as the use of new buzz words, old slang expressions and idiolectal formations created on the spot in order to meet aesthetic-artistic expectations [3] as well as intrinsic metric restrictions of the genre of rap music [6].

3 Corpus Priorities and Choices

The creation of a linguistic corpus of francophone rap songs began in spring 2009, when the first one hundred rap songs were included in the Bonito corpus manager[2] [14]. That is also when the corpus obtained its name and logo[3]. The original idea was to create a corpus based on rap music production in

[1] The first data compiling and text processing were accomplished thanks to the enthusiastic work of our student Jiří Marek.

[2] Thanks to the web page description at `https://nlp.fi.muni.cz/projects/bonito/` (still active).

[3] For additionnal presentation, see: `https://is.muni.cz/do/phil/Pracoviste/URJL/rapcor/index_en.html`.

France but students quickly became interested in extending the text compilation by involving other francophone hip-hop scenes as well (i.e., French-speaking rappers from Belgium, Switzerland, Canada or even from Senegal including French overseas territories and departments, e.g. Reunion Island in the Indian Ocean). Furthermore, another one hundred texts from Czech and Slovak rap albums were prepared for tokenization but were not published yet.

So far, we have relied on transcriptions of lyrics that the authors themselves insert into album booklets. Until today, we have been listing in our Google document database 2,424 francophone rap albums (EP, LP and mixtapes) published between 1984 and 2020, with 904 (37 % of whole database) physically verified against their CD or vinyl versions. In sum, almost 53 % (479) of all verified booklets contain at least one text transcription, in 16 % of them (78) all texts are transcribed. The advantage of this method (as opposed to obtaining text transcripts from fan pages on the internet) is that it deals with authors' licensed texts, thus enabling us to identify their own grammar and typing errors, as well as any disparities between the written- and the sung-versions (labelled as P and S versions in RapCor's database). Such differences are often intentional, particularly where the sung-version (S) is way too explicit. Another advantage of this approach is that it helps us to obtain a huge variability of transcriptions of oral neologisms that do not have a fixed graphic form yet (due to the phonemic orthography of written French, in comparison to Czech, for example). Moreover, this method allows us to compare P and S versions in parallel-corpora, using *MK-align* software[4] (see the scholar work of Vaňková, 2014, listed in the next link) and to point out the most frequent categories of those P and S disparities (consisting mainly in non-transcription of introductive or final rappers' "gimmicks", territorial references and echoed voices).

The newest version of RapCor, released on 21st October 2020 and available in the Sketch Engine corpus manager, is *RapCor 1288*, i.e. having the lyrics of 1288 songs. Relying on authors' transcripts still remains the most original feature of RapCor's text processing, even though this corpus version does contain 133 texts of songs by well-known rappers that were gathered from internet's fan pages because the rappers never transcribed their lyrics into album booklets. Their inclusion is the result of our students' decision, either motivated by seeking a representative nature of their sub-corpora in several theses[5] or by their personal choice for seminar projects on lyrics translation into Czech. In the future, we expect that the number of unauthorized lyrics retrieved from the internet (or authorized by interprets directly on the internet) will increase because the importance of publishing in physical format is decreasing.

As the building of the RapCor corpus is strictly based on students' work, and as our students are mostly non-native speakers of French, it is not possible to ask them for their own transcriptions of the voice on the music track. Rappers often fill their "punchlines" with substandard and trendy words, including

[4] Freely available at: `http://www.tal.univ-paris3.fr/mkAlign/#p1` (see [2]).

[5] Listed at: `https://is.muni.cz/do/phil/Pracoviste/URJL/rapcor/kvalifikacni_MU_en.html`.

borrowings from non-European languages that reflect the dynamics of lexical innovations in multicultural suburbs. The frequency of the "XXX" symbol (i.e. a word or words that is impossible to hear or understand) in online published version of corpus (301 times) serves as evidence of how difficult it is when one tries to accurately transcribe the rappers' rappings.

The frequent gaps in lyrics transcriptions on the internet clearly show that speech recognition is sometimes hard even for native speakers and experts in local hip-hop scenes. That is one of the reasons why the digital media company *Genius.com* was so successful in 2014 with their song annotation system. The company changed their earlier "explanations" of hip-hop lyrics on their original website *RapGenius.com* into a collaborative database, based on the contributions of "annotators" of lyrics and opened itself to other music genres.

Online publication of rap lyrics in French started on RapGenius just in 2010, but until the re-launch of Genius, our interest in searching for non-authorized text transcripts has led us to explore several other web sites. Thanks to our collaboration with The Natural Language Processing Centre at the Faculty of Informatics, a more convenient extraction of texts from various specialized websites (so-called *RapCor Text Crawler*[6]), was launched in 2016. With this tool, online versions of lyrics, if available, can be obtained in several (from one to six) different versions, which helps our students to clarify content ambiguities in case of doubt over the accuracy of the fan's transcription of the analyzed song. Together with booklet lyrics, there are currently (November 2020) 4,588 transcriptions of francophone rap songs in different stages of corpus treatment.

4 Text Processing Methods

The preparation of texts to include in the linguistic corpus is much more time consuming in case of authentic lyrics' taken from album booklets than lyrics taken from internet text crawlers. For that reason, the expansion of RapCor has been relatively slow in comparison with other text corpora, mainly those crawled from web pages with automatic annotations[7].

Firstly, students are completing structural metadata about songs (singer, featuring, album, publication year, song's length) and, more recently but not exhaustively, open personal metadata about artists (date and place of birth, geo-localisation, parents' origins, etc.).

Prior to tokenisation, there are several text processing phases:

1. the scanning of texts from booklets (from 2015 until now, 684 albums were (re)scanned in high resolution);

[6] Thanks to Pavel Rychlý's student Lukáš Banič, available at: `https://nlp.fi.muni.cz/projekty/rapcor/`.

[7] By writing an open-source script with R program in order to crawl texts from Genius.com, Corentin Roquebert [13] offered in 2015 a comprehensive to double the size of RapCor in a quite short time (see `https://nycthemere.hypotheses.org/541`). For various discourse analysis, this quantitative method is sufficient but it doesn't offer further qualitative features as our text processing methodology.

2. after encoding of available transcriptions of songs, the texts are cut and vertically pasted (presently 2,427 documents);
3. the result of automatic text resolution (OCR) is then checked and saved as text recognized image (decision made just in 2015; until now 664 texts were saved as double-layer pdf files in our internal storage);
4. the machine-recognized and word-by-word controlled text is tagged by structural information tags (full title, interpret, featuring, album, year of publication, length of music track) and two text files, P (written-) and S (sung-) versions, are created. The authorized booklet text's transcription (P version) is checked against audio recordings (S version) and any differences found in lyrics are colour-coded according to predefined disparity categories (both in P and S version). As of now, 2,323 P versions have been prepared by controlling OCR recognition and by adding structural tagging. Because of the time-consuming overhearing of audios, the mirror S version has already been completed for only 1,332 of them. There are actually seven categories of disparities to be coloured:

   – correction of typographical or grammatical error;
   – deliberately missed correction if it was author's intention (mainly for jokes or special adaptations of loanwords);
   – replacing of text by another one (frequent for vulgarisms);
   – addition of text (especially opening and closing sections called "intro" and "outro" are sung but not transcribed in booklets), including punctuation in order to increase the quality of morphosyntactic annotations (lyrics are often written with upper case in the beginning of each punchline and without any finishing mark of sentences);
   – omission of text (less frequent but annotated as empty token);
   – different position of stanza or chorus in booklet transcription in comparison of what is really sung;
   – pronunciation mismatch (frequent for graphically non-adapted loanwords; annotation include phonetic transcription in international phonetic alphabet);

   and the a special category with colours belongs to the aforementioned "XXX" in S versions of all types:

   – incomprehensible passage (i.e. waiting for overhearing by native speakers).

   Only after creating of S version of two types (a) colour-coded files in case of traditional text processing as explained before or b) just tagged by structural information tags in case of fans transcriptions from the internet – other 142 texts), we can proceed to the final stage of pre-annotation text processing;
5. the creation of a plain text without any tags that can be submitted for automatic tokenization and semi-automatic annotation (1,474 texts are completed until this stage).

5 Remarks on Lemmatisation and Annotation

When the first texts for corpus were prepared, there was only one tool available as open-source, the famous *TreeTagger* tool[8]. For this reason, all segmentation and annotation was carried out by the TreeTagger, even if the result had to be reviewed carefully because of its poor dictionary in the matter of oral/substandard French. About 2012, we also tested some RapCor texts on a licensed tool *Cordial Analyseur*[9] with more successful syntactic annotation results but its wider use became unsustainable in our collaborative project for financial and technical reasons.

Until 2019, students were adding words that TreeTagger lemmatized as "unknown" to the shared Google table file. This table was designed as an internal dictionary with a special focus on loanwords, inversed slang words (well-known as "verlan"), regionalisms (mainly from well-represented Canadian French) and socio-cultural references invoked by usage of proper names (mainly toponyms and anthroponyms). Thanks to close collaboration with Pavel Rychlý during this year, students don't have to download and learn to handle TreeTagger tool in their computers and then search for unknown lemmas in our dictionary anymore. By simple pasting of the plain text into a so called French tagger web page window, together with the code of the song[10], they can download encoded Open office table file with tokenized and automatically pre-annotated text. The special feature is adding to TreeTagger's lemmatization result lemmas from our dictionary. Furthermore, tokens lines containing unguessed ambiguities, disparities between both dictionaries and unknown words are coloured in three different colours in order to attract students' attention. This will help us to decrease the number of errors in a decision process for POS and lemmas. The advent of big web corpora enables us to verify frequencies in writing down of orality.

As the corpus building started a long time before that, the question of unique lemma's choice in case of several graphic variants was solved by applying of so called *method of lexicographic filters* [9:341-353]. For example, [tɛs] is an identitary neologism which knew a quick diffusion in common youth slang and it is semantic equivalent of the aforementioned word *cité* (created by truncation of bissyllabic (verlan) inversion: *cité* [site] > *téci* [tesi] > [tɛs]. In booklets, rappers transcribed this word as *tess, tece, téc, té-ce(s), tèc* and *tèce*. As it is absent from reference dictionaries such as *Le Petit Robert* or *Le Petit Larousse*, the search of lemma passed through the first lexicographic filter and, as explained in [16:154], the lemma *téc* was fixed for our corpus, according to the graphic form chosen as entry in unique slang dictionary with academic background that recorded [tɛs] at that time (Goudaillier's 3rd edition of *Comment tu tchatches!*, published in

[8] Freely available at: `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/` (for detail explanation in French, see [19]).

[9] Called also Cordial Universités, this product of French company Synapse development was widely used in lexicometric studies [7].

[10] Available at: `https://nlp.fi.muni.cz/projekty/rapcor/tag.cgi`.

2001)[11]. However, the aging of the then trendy words brings usually an increase of graphic match that can be seen on frequencies and year stamp of each form: *tece* (once in a song released in 2004), *téc* (once in 2009), *tè-ce* and *tè-ces* (both once by the same singer in 2005), *tèce* (twice in 2004 and 2006) and *tèc* (twice in RapCor, in 1998 and 2009), in comparison of longitudinal use of *tess* (48 times in RapCor1288). This form dominates also in very large web corpora for French (both in *FrTenTen* 2012 and 2017 or in *Araneum Francogallicum*) where it is mixed with homonymic female name *Tess* (written often without upper case). While occurrences in those big corpora exceed easily several hundreds, our small numbers are more relevant statistically (e.g. relative frequency of *tess* in RapCor is 62.54 i.p.m, but only 0.71 i.p.m. in FrTenTen12, which is still more than 0.5 i.p.m. in newer FrTenTen17). This example shows the dynamics of slang words in recent diachrony. It is also a typical case of our "drifting in the current", when corpus building includes permanent integrating of new NLP tools on old and new lyrics and (re)considering lemmatization for old and new slang words.

6 Conclusion and Future Work

The RapCor corpus is to our knowledge the only existing comprehensive linguistic corpus of francophone rap songs, which were released between 1984 and 2020. In its current published version, RapCor incorporates a sample of 1,288 individual songs from 169 albums and sung by 570 artists. However, this is still a relatively small part from the universe of the French rap production and a small portion of rappers' population. In order to be able to benefit from the full potential of the database (another 3,300 songs are under treatment and thousands of others can be incorporated already), RapCor needs further expansion and equilibration. Together with displaying of P and S versions disparities in the next version, our short-term objective is to rebuild our dictionary and to test other taggers than TreeTagger, especially UDPipe and FreeLing.

Acknowledgments. This work was supported by the project of specific research "Románské jazyky a románské literatury 2020" (Romance languages and romance literatures 2020), project no. MUNI/A/1262/2019.

References

1. Cartier, E. (2017). *Neoveille, a Web Platform for Neologism Tracking.* Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017. `https://www.aclweb.org/anthology/E/E17/E17-3024.pdf`.

[11] This choice of *téc* is logic if one wants to refer to original word *téci* but improper form phonological point of view (*é* refers always to closed-mid vowel [e] and not to open-mid [ɛ]). This confirms the Tengour's choice of the entry *tèce* (with sub-entry *tess*) for its online peri-urban slang dictionary (`http://www.dictionnairedelazone.fr`; printed version was published in 2013).

2. Fleury, S. (2012). *MkAlign (version 2.0) : Manuel d'utilisation.* Paris: Université Sorbonne Nouvelle Paris 3.
3. Ghio, B. (2012). *Le rap français: désirs et effets d'inscription littéraire. Disertation Thesis.* Paris: Université Sorbonne Nouvelle – Paris 3.
4. Goudaillier, J.-P. (2019). *Comment tu tchatches! Dictionnaire du français contemporain des cités.* 4th edition. Paris: Maisonneuve & Larose, (1st ed. 1997).
5. Hammou, K. (2012). *Une histoire du rap en France.* Paris: La Découverte.
6. Chodakova, P. (2014). *Metrická inovace ve francouzském a českém rapu.* Lingvistika Praha. http://lingvistikapraha.ff.cuni.cz/node/199.
7. Lebart, L., Pincemin, B. and Poudat, C. (2019). *Analyse des données textuelles.* Québec: Presses de l'Université du Québec.
8. Paryzek, P. (2008). *Comparison of selected methods for the retrieval of neologisms.* Investigationes linguisticae, Vol. XVI, pp. 163–181.
9. Podhorná-Polická, A. (2009). *Universaux argotiques des jeunes.* Brno : Munipress.
10. Podhorná-Polická, A. and Fiévet, A.-C. (2009). *À la recherche de la circulation d'un néologisme identitaire: le cas de bolos.* In Kacprzak, A. and Goudaillier, J.-P. (eds.). Standard et périphéries de la langue. Łódź: Oficyna Wydawnicza Leksem, pp. 207–223.
11. Polická, A. (2018). *Lexikální inovace. Dynamika šíření identitárních neologismů.* Brno: Masarykova univerzita. `https://www.muni.cz/inet-doc/1129409HAB_SPIS_Policka_2018_final.pdf`.
12. Renouf, A. (2016). *Big data and its consequences for neology.* Neologica, 10, pp. 15–38.
13. Roquebert, C. (2015). *Tutoriel: Récupérer des paroles de rap du site Rapgenius* [online]. Nycthémères. Mesures du rap. Academic blog Hypotheses.org. `https://nycthemere.hypotheses.org/533`.
14. Rychlý, P. (2007). *Manatee/Bonito – A Modular Corpus Manager.* 1st Workshop on Recent Advances in Slavonic Natural Language Processing, pp. 65–70.
15. s.a. (2012). *Pas de blème ! Slovník slangu a hovorové francouzštiny.* Brno: Lingea (1st edition 2009).
16. Sekaninová, T. (2012). *Stéréotypes liés au verlan : variation diatopique dans le rap français* [master's thesis]. Brno: Masaryk University. `http://is.muni.cz/th/263203/ff_m/`.
17. Sláma, J. (2017). *K (polo)automatické excerpci neologismů*. Jazykovědné aktuality, LIV, 3–4, pp. 34–46.
18. Spady, J., Meghelli, S. and Alim, H. S. (2006). *Tha global cipha: Hip Hop culture and consciousness.* Philadelphia: Black History Museum Press.
19. Stein, A. and Schmid, H. (1995). *Étiquetage morphologique de textes français avec un arbre de décisions.* Traitement automatique des langues, 36, 1-2 (Traitements probabilistes et corpus), pp. 23 –35.
20. Tengour, A. (2013). *Tout l'argot des banlieues. Le dictionnaire de la zone en 2 600 définitions.* Paris: Les éditions de l'Opportun.

Semantic Analysis of Russian Prepositional Constructions

Victor Zakharov[1], Kirill Boyarsky[2], Anastasia Golovina[1], and Anastasia Kozlova[2]

[1] Saint Petersburg University
Universitetskaya emb. 7-9
199034 Saint Petersburg, Russia
v.zakharov@spbu.ru, st070508@student.spbu.ru
[2] ITMO University
Kronverkskiy av. 49
197101 Saint Petersburg, Russia
boyarin9@yandex.ru, stasia.kozlova@gmail.com

Abstract. The paper deals with semantics of Russian prepositions. We consider prepositional meaning to be the relation found in prepositional constructions where it should be regarded as a special type of relationship between content words. There is a rather small set of common prepositional meanings that we call syntaxemes after G.A. Zolotova that encompass the larger part of the Russian prepositional semantics as a whole. In this paper we propose a methodology of prepositional phrase extraction and syntaxeme identification based on text corpora and corpus statistics. Prepositional construction meaning is detected automatically using the SemSin parser. We demonstrate our methodology on constructions with the polysemous Russian preposition *через* (through).

Keywords: Russian prepositional constructions, preposition meaning, corpus statistics, parsing, semantic classes

1 Introduction

This paper presents a study on semantics of Russian prepositions. It is part of a larger project. In this paper, we demonstrate the way to identify preposition meaning on constructions with the polysemous Russian preposition *через*.

The preposition is a part of speech found in many languages. Russian linguistics divides this class into primary and secondary prepositions by origin as well as simple (one word) and complex (multiword) units by structure. Primary prepositions in particular are highly polysemous. For instance, the Russian preposition 'with' has 26 meanings in the Dictionary of the Russian Language [4] (11 meanings with the genitive case, 2 with the accusative one and 13 with the ablative). The majority of them are quite rare, in some cases the preposition is a part of an idiom.

Prepositional ambiguity is manifested in the complex nature of the prepositional meaning and in selective preferences of certain prepositions, depending

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 103–111, 2020.

on context. That alone makes the systematization of the prepositional class a very complicated and tedious undertaking.

Prepositions are often regarded as having no lexical meaning. However, we have an alternative view on prepositional semantics. We consider prepositional meaning to be the relation found in prepositional constructions where it should be regarded as a special type of relationship between content words. An additional factor in the proposed view on the prepositional meaning is the case of the prepositional governee. We believe that the preposition should be studied in conjunction with the associated case. It becomes possible, then, to speak of prepositional homonymy where every "preposition+case" pair is a homonym within the paradigm of a given preposition.

We have adopted the approach suggested by G.A. Zolotova [10] for the task of describing prepositional meanings. The preposition-case unit is regarded as a syntaxeme – the minimal lexical-grammatical construction expressing a certain meaning. Syntaxemes can be relatively autonomous, but usually they form blocks which attach themselves to notion words, mostly verbs. There are about 30 syntaxemes listed in Zolotova's dictionary. Different preposition-case units may form semantically comparable syntaxemes, which is why in this study we take the syntaxeme to mean the common semantic invariant of these units. The names of some syntaxemes (temporative, directive, instrumentive, etc.) correlate with the idea of syntactic-semantic roles that have been introduced by Ch. Fillmore with the idea of syntactic-semantic sentence description [5].

We also use the concept of the syntaxeme as a node in prepositional ontology. The notion of the syntaxeme was defined in the functional direction of traditional linguistic analysis, so we redefine it inside our own quantitative corpus approach [1].

In contrast to the classical linguistics focusing on the simplest units of different language levels, modern studies practice synthetic methods attempting to capture and describe the more complex language structures which integrate different language units: words, collocations, etc. In classical linguistic papers, prepositional constructions used to be described from the grammatical point of view and their semantics used to be neglected. Complex description and systematization of prepositional constructions demand elaboration of identification methods using manual and automatic techniques as well as analysis of their paradigmatic and syntagmatic features and quantitative analysis of their frequency and strength.

2 Russian Preposition *Через*

This article presents the methodology of semantic analysis of prepositions based on constructions with the Russian preposition *через* ('through, across, in, after').

Existing approaches to prepositional semantics description differ in their methodology, both formally and in their content. Deep research on the semantics of individual prepositions, including *через*, can be found in linguistic literature. Such is, for instance, the specialized comparative analysis of the seman-

tics of *через* and its close synonym *сквозь* ('through') by V.A. Plungyan and E.V. Rakhilina [8]. Their paper presents an investigation on the semantics of these two prepositions based on their various aspects. Full descriptions of polysemous prepositions are provided in the form of semantic nets consisting of blocks of ready-made and constructible language material, including idioms. It can be said, however, that the description is provided in the terms of the construction-based approach practiced by us. Also introduced are the notions of "stable" and "flexible" word parameters which are descriptions of situations gained through inferring semantic characteristics of words in context.

However, most sources operate with a significantly simpler system of meanings. Wiktionary, for instance, lists 4 meanings of the preposition *через*:

1. *сквозь, поперёк* ◆ *Он помог слепой женщине перейти* **через** *дорогу на другую сторону.*
2. *поверх чего-либо* ◆ *Я легко перепрыгнул* **через** *забор.*
3. *по истечении некоего отрезка времени* ◆ *Перезвони мне минут* **через** *пять.*
4. *с помощью, посредством* ◆ *Оплата производится на почте при получении заказа,* **через** *Сбербанк либо по WebMoney.*

As can be seen from this example, the meanings of prepositions in explanatory dictionaries are typically expressed descriptively or by means of other synonyms, forming a "vicious circle". However, the same set of meanings could be interpreted as the transitive (1, 2), temporative (3) and mediative (4) syntaxemes as per the Syntactic Dictionary by G.A. Zolotova [10].

The **transitive** syntaxeme is one of the possible ways of the proposition localization. Unlike the characteristics of location, which are applicable to a diverse set of actions, states and processes, this specification is often associated with the "framework" structure of the prefix *пере-* for the verbs of motion and their derivatives: *перейти через дорогу* 'to go across the road', *перевозки нефти через Атлантику* 'oil transits across the Atlantic', etc.

The **temporative** rubric is quite diverse. In some cases, the time specification appears to be relative: the time interval precedes some event, follows it or is simultaneous with it. Another variation conveys a sequence of events, which can be expressed with the preposition *через* 'in, after' with the accusative case: *прийти через день* ('to come in a day'), *произойти через 2 столетия* ('to happen after 2 centuries'). However, the corresponding Wiktionary definition of this meaning may need to be reformulated or divided into two, the temporative and the locative, as there exists a very similar meaning of *через* that refers to alternation in space: *деревья высаживают через 1.5 м в ряду* 'the trees are planted every 1.5 m in a row', *по всей длине не реже чем через два метра* 'along the entire length no sparser than every two meters'.

The **mediative** as a semantic rubric has a narrow and a wide interpretation. Generally, it is regarded as a particular semantic role in the predicate structure of a verb. In the narrow sense the mediative is understood as a means, that is, a substance or an object used during the performance of an action or a process. In a broader sense the mediative is a tool (the instrumentive meaning) and includes

its material and abstract implementations [7]. In the Russian language both the mediative and the instrumentative are regularly expressed by the instrumental case form (*красить стены валиком* 'to paint walls with a paint roller', *рисовать картину красками* 'to paint a picture with paints'), however, we can observe more complex syncretic instances in the form of prepositional constructions.

Other syntaxemes of this preposition are rarely observed.

3 Methodology of Corpus Statistical Analysis

We have developed a procedure for describing the continuum of prepositional meanings basing on the corpus data starting from the bottom – that is, textual analysis of sense distribution in random context samples from different corpora. The stages of our procedure are as follows:

- Acquisition of sets of prepositional constructions from corpora of different types and different functional styles;
- Acquisition of a number of statistical characteristics for each preposition from corpora of different types and functional styles, namely:
  - ipm in a corpus (corpora);
  - percentage of each meaning of appropriate preposition;
  - a list of most frequent semantic classes and/or lexemes acting as a "governor" for each prepositional meaning;
  - a list of most frequent semantic classes and/or lexemes acting as a "governee" for each prepositional meaning.

The semantic-grammatical analysis of relations between lexical items certainly cannot be performed entirely automatically and requires participation of linguists. To estimate the percentage of each meaning of the preposition *через* we have resorted to expert evaluation of the selected constructions (contexts). Those were annotated according to the meaning realised in the given context, for example:

- *он наблюдал бы прохождение Юпитера* **через** *диск Солнца* → transitive;
- **через** *минуту дверь открылась* → temporative;
- **через** *богослужение мы действительно достигаем истины* → mediative.

The Russian National Corpus (RNC) was used as the base material source.

We found that out of the 5 meanings ascribed to the preposition *через* by G.A. Zolotova [10] only 3 appear to be present in real texts (Fig. 1).

All of the observations suggest that the bottom-up corpus-based approach is imperative in the task of studying preposition semantics. However, the context window method used originally in the study was discovered to be insufficiently effective as the actual governor and governee, which are crucial in prepositional phrase identification, are not always captured by the window. The quality of automatically extracted prepositional constructions could be improved through the use of full syntax parsing. Furthermore, it is impossible to process and annotate large text arrays without using reliable automatic analysis tools.

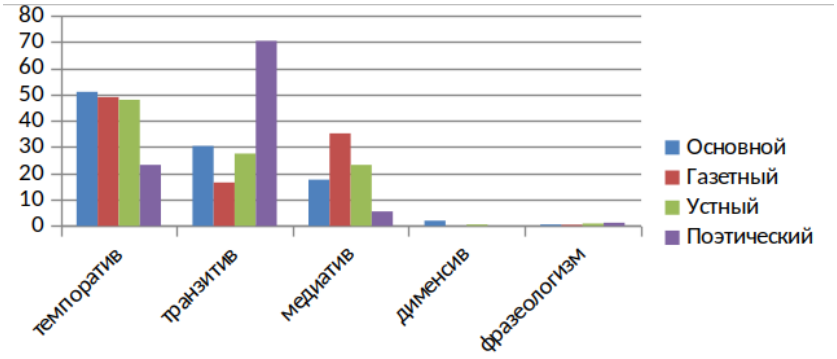


Fig. 1. Ratios of *через* meanings in RNC subcorpora

4 Automatic Parsing of Prepositional Phrases with *Через*

Unfortunately, the choice of openly available tools capable of performing a detailed semantic analysis is very limited for Russian. Identification of prepositional phrase semantics is one of the most difficult tasks for a parser due to some specific features of prepositions. Firstly, prepositions are not declinable, and many consist of just one or two letters. That makes it impossible to rely on the morpheme (root or word ending) during the analysis. Secondly, a high degree of governee homonymy does not allow for unambiguous semantics identification through the analysis of just the prepositional phrase and often demands a full sentence analysis.

In order to study semantic meanings of prepositions we used the SemSin parser [3], which builds a dependency tree for each sentence and detects types of relations between its nodes. The parser relies on a semantic-syntactic dictionary and a classifier, both of which are extensions of the semantic dictionary by V.A. Tuzov [9]. The dictionary currently contains about 200 000 lexemes belonging to 1700 classes. An important element of the system dictionary is a table of prepositions containing over 2200 combinations of semantic categories of nouns with which prepositions can interact as well as the names of relations between governors and prepositional phrases.

Each sentence undergoes morphological analysis involving tokenization, after which the lexical analyser transforms the linear sequence of tokens into a dependency tree with the help of a system of production rules [2].

In order to test the accuracy of the automatic semantics identification we have used the parser to analyse corpora of around 100 000 tokens each of texts representing different functional styles: newspaper and magazine articles, fiction, scientific papers and texts, legal documents and oral speech transcripts. All the sentences containing the preposition *через* (50 to 100 depending on the corpus) were then selected for further inspection. The correctness of prepositional phrase governor and governee detection was checked by experts. In some

complex cases parsing results were checked against those made by the ETAP-4 parser [6].

The following sentence is an example from the Russian National Corpus: *Не хочется, чтобы через определённый промежуток времени у нашей молодёжи настолько поменялись приоритеты* 'It would be undesirable if the priorities of our youth changed that much in a certain span of time'. The word *промежуток* 'span' has two semantic meanings: temporal and locative, which is reflected in the RNC annotation. In the analysis of this sentence with ETAP-4 the prepositional phrase is linked to the governor, the verb *поменялись* 'changed', through the "adverbial adjunct" relation, which does not help to resolve the semantic homonymy. However, the *Когда-relation* ('When') of the tree built by SemSin (Fig. 2) unambiguously detects the meaning of the preposition as temporative and allows for the temporal interpretation of the word *промежуток* 'span' only.

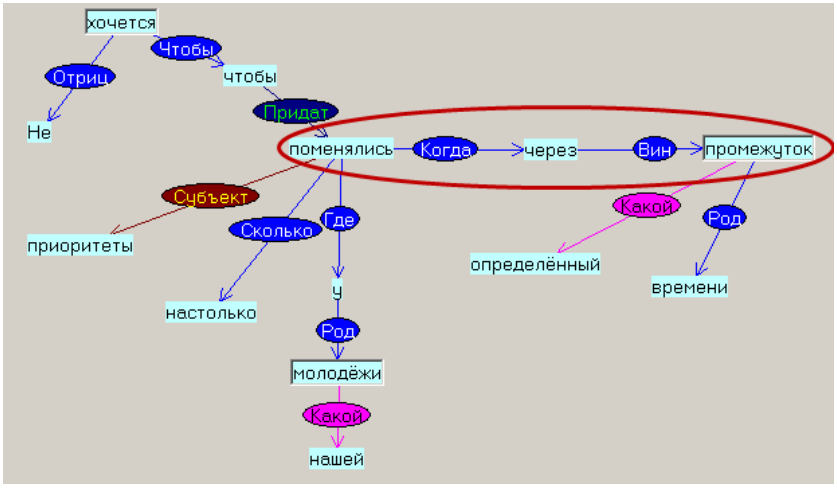


Fig. 2. The preposition *через* in the temopative meaning

Cases in which the preposition is found at a significant distance from its governor or governee present a considerable challenge for automatic parsing. This is especially characteristic of legal texts, such as laws, statutes, etc. For instance, in the following sentence fragment (full sentence length: 71 words) *при наличии... необходимого оборудования* **передавать** *в соответствии со стандартными процедурами Всемирной метеорологической организации в основные международные синоптические сроки* **через** *береговой* **радиоцентр**… *оперативные данные...* 'in the presence of… necessary equipment **transmit**… live data in accordance with the standard procedures of the World Meteorological Organization in the main international synoptic hours **through** the coastal **radio centre**' the preposition *через* is located 14 words away from its governor *передавать* 'transmit'. ETAP-4 wrongly links the prepositional group to the

word *сроки* 'hours', while SemSin correctly detects the true governor (Fig. 3). The fact that the governee *радиоцентр* 'radio centre' belongs to the semantic class of establishments allows to infer that the preposition has the mediative meaning in this context.

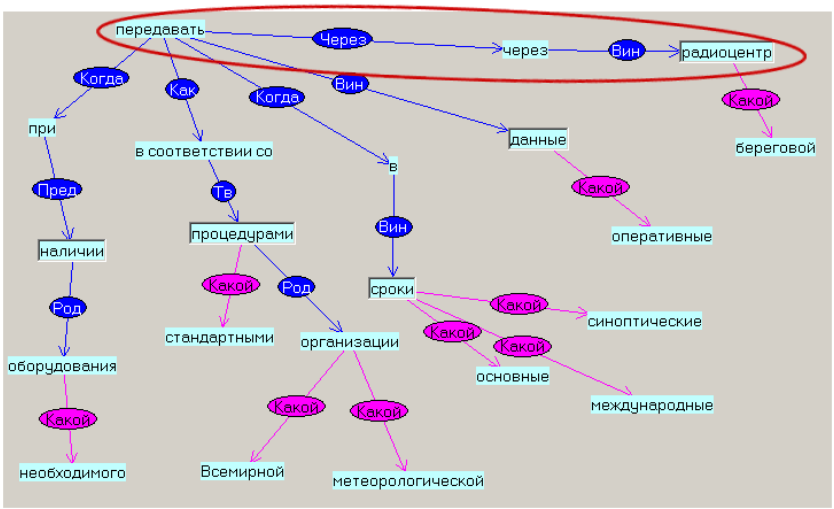


Fig. 3. The preposition *через* in the mediative meaning

Although the maximum distance from a preposition to its governee is much shorter than to its governor, issues do occur in some cases, especially when there are punctuation marks between the preposition and its governee, like in the case of parenthetical phrases marked off by commas, e.g. *Дина понуро шла через широкий, как площадь, двор гаража* 'Dina was walking gloomily through the wide as a square court'. ETAP-4 considers the governee to be the word *широкий* 'wide'. SemSin locates the governee *двор* 'court' correctly while also resolving the semantic homonymy of the word *двор* (plot of land vs. social category). Therefore, the semantics of the preposition is identified as transitive.

The results of the accuracy evaluation of the automatic formation of prepositional phrases with the preposition *через* and the detection of the relation type, e.g. the semantics, by the parser are provided in Table 1.

Thus, we can conclude that the SemSin parser provides prepositional phrase semantics detection of sufficient quality. That being said, the accuracy could be improved by means of improving parsing rules and detalization of interactions between prepositions and nouns of various semantic classes.

One of the objects of our current research is the preposition *с* ('with', 'from'). The phase requiring the most detailed examination is the extraction of syntaxemes, which are listed below (Table 2).

Table 1. Accuracy of formation of prepositional phrases with preposition *через* by the parser

Text type	Governor/governee accuracy	Relation type accuracy
Oral transcripts	87 %	87 %
Newspaper articles	87 %	85 %
Scientific texts	92 %	73 %
Fiction	92 %	87 %
Legislative texts	65 %	78 %

Table 2. Syntaxemes of the preposition *с* ('with', 'from')

Syntaxeme	Examples
Directive	*съёмка со спутников* 'images from satellites'
Instrumentive	*кормить с ложки* 'to feed from a spoon'
Source	*перевод с китайского* 'translation from Chinese'
Object	*помочь с задачей* 'to help with a problem'
Comitative	*пирог с начинкой* 'pie with a filling'
Cause	*закричал с радости* 'cried for joy'
Comparison	*длиной с полметра* 'half a meter long'

The least frequently occurring syntaxemes of the preposition *с* are "cause" and "comparison", which, in combination with their syntactic complicacy, makes them the most difficult for parser identification. Still, with some of the grammatical particularities determined, it is highly possible to bring the level of automatic identification above the current threshold.

5 Conclusion and Future Work

All of the observations presented in the paper suggest that our approach is suitable for use in the task of studying preposition semantics. Further stages of our research include expansion of the application of the methodology presented in this paper to other prepositions.

Additionally, research on the prepositional use in fixed phrases and idioms has been started.

To improve the quality of extracted constructions and to reduce human participation and labor costs a syntactic parser operating on a base of semantic categories is to be used in further studies. We strive for automatic identification of preposition meanings as demonstrated in the current paper on the preposition *через*.

The conducted research shows that the SemSin parser successfully finds prepositional groups with the preposition *через* as well as others and determines the type of semantic connection with a high degree of accuracy. However, to automatically determine the semantics of prepositions that have a greater variety of semantic meanings, additional research is needed on the compatibility of prepositions with nouns.

Acknowledgements. This work has been supported by the Russian Foundation for Basic Research, project No. 17-29-09159.

We express our heartfelt gratitude to Irina Azarova, Andrei Masevich and Anna Moskvina for their help in processing corpus materials.

References

1. Azarova, I., Zakharov, V., Khokhlova, M., Petkevič, V.: Ontological description of Russian prepositions. In: CEUR Workshop Proceedings, 2552, pp. 245-257 (2020).
2. Boyarsky, K., Kanevsky, Ye. A system of production rules for the building of a sentence syntax tree [Sistema produkcionnykh pravil dlya postroyeniya cintaksicheskogo dereva predlozheniya]. In: Applied linguistics and linguistic technology [Prikladna lingviskika ta lingvistichni tekhnologii], MegaLing-2011. Kyiv, Dovira, pp. 73-80 (2012).
3. Boyarsky, K., Kanevsky, Ye. Semantic-syntactic parser SemSin [Semantiko-sintaksicheskiy parser SemSin]. In: Scientific and technical bulletin of information technologies, mechanics and optics [Nauchno-tekhnicheskiy vestnik informacionnykh tekhnologiy, mekhaniki i optiki]. Vol.15. No. 5, pp. 869-876 (2015).
4. Dictionary of the Russian language: in 4 volumes / Russian Academy of Sciences, Institute of Linguistic Research. 4th edition, stereotyped. Moscow: Russian language: Polygraph resources (1999)
5. Fillmore, Ch. J. The Case for case. In: Universals in Linguistic Theory. Bach and Harms (Ed.). New York: Holt, Rinehart, and Winston, pp. 1-88 (1968).
6. Language processor ETAP-4 [Lingvisticheskiy protsessor ETAP-4]. URL: `http://proling.iitp.ru/ru/etap4` (last access: 28.10.2020).
7. Mustajoki, A. Theory of functional syntax [Teorija funtsionalnogo sintaksisa]. Moscow (2006).
8. Plungyan, V.A., Rakhilina, E.V. Function word polysemy: the prepositions *через* and *сквозь* [Polisemiya sluzhebnykh slov: predlogi cherez i skvoz]. In: Russian studies today [Rusistika segodnya]. No. 3, pp. 1–17 (1996).
9. Tuzov, V.A. Computational semantics of the Russian language. Saint Petersburg: SpbU publishing (2004).
10. Zolotova, G.A.: Syntactic Dictionary: a set of elementary units of the Russian syntax [Sintaksicheskiy slovar': repertuar elementarnykh edinits russkogo sintaksisa]. 4th edition. Moscow (2011).

Removing Spam from Web Corpora Through Supervised Learning and Semi-manual Classification of Web Sites

Vít Suchomel[1,2]

[1] Natural Language Processing Centre
Masaryk University, Brno, Czech Republic
xsuchom2@fi.muni.cz
https://nlp.fi.muni.cz/en/
[2] Lexical Computing, Brno, Czech Republic

Abstract. Internet spam is a major issue hindering the usefulness of web corpora. Unlike traditional text corpora collected from trustworthy sources, the content of web based corpora has to be cleaned.
In this paper, two experiments of non-text removal based on supervised learning are presented. First, an improvement of corpus based language analyses of selected words achieved by a supervised classifier is shown on an English web corpus. Then, a semi-manual approach of obtaining samples of non-text web pages in Estonian is introduced. This strategy makes the supervised learning process more efficient.
The result spam classifiers are tuned for high recall at the cost of precision to remove as much non-text as possible. The evaluation shows the classifiers reached the recall of 71 % and 97 % for English and Estonian web corpus, respectively.
A technique for avoiding spammed web sites by measuring the distance of web pages from trustworthy sites is studied too.

Keywords: Web corpora, Web spam, Supervised learning.

1 Introduction

It is known that boilerplate, duplicates, and spam skew corpus based analyses and therefore have to be dealt with. While the first two issues have been successfully addressed, e.g. by [8,10,17,12], spam might be still observed in web corpora as reported by [7,14].

While the traditional definition of web spam is *actions intended to mislead search engines into ranking some pages higher than they deserve* [4], the text corpus point of view is not concerned with intentions of spam producers or the justification of the search engine optimisation of a web page. A text corpus built for NLP or linguistics purpose should contain coherent and consistent, meaningful, natural and authentic sentences in the target language. Only texts created by spamming techniques breaking those properties should be detected and avoided.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 113–123, 2020.

The unwanted non-text 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. stock market reports, weather forecast, sport results – all of the same kind very similar), and finally any incoherent text. Varieties of spam removable by existing tools, e.g. duplicate content, link farms (quite a lot of links with scarce text), are only a minor problem.

Automatically generated content does not provide examples of authentic use of a natural language. Nonsense, incoherent or any unnatural texts such as the following short instance have to be removed from a good quality web corpus: *Edmonton Oilers rallied towards get over the Montreal Canadiens 4-3 upon Thursday.Ryan Nugent-Hopkins completed with 2 aims, together with the match-tying rating with 25 seconds remaining within just legislation*.[3]

Avoiding web spam by selecting trustworthy corpus sources such as Wikipedia, news sites, government and academic webs works well: [2] show it is possible to construct medium sized corpora from URL whitelists and web catalogues. [13] used a similar way of building a Czech web corpus. Also the BootCaT method [3] indirectly avoids spam by relying on a search engine to find non-spam data. Despite the avoiding methods being successful yet not perfect [14], it is doubtful a huge web collection can be obtained just from trustworthy sources.

Furthermore, language independent methods of combating spam might be of use. [9] reported web spamming was not only a matter of the English part of internet. Spam was found in their French, German, Japanese and Chinese documents as well. According to our experience in building web corpora in more than 50 languages, non-text content is still on the rise.

In this paper, two experiments of spam removal based on supervised learning are introduced: Section 2 shows the improvement of corpus based language analyses of selected words achieved by a supervised classifier applied to an English web corpus. Section 3 presents an experiment with an Estonian web corpus. A semi-manual approach of obtaining samples of non-text web pages made the supervised learning process more efficient. The result spam classifier reached a very high recall of 97 %. The usefulness of measuring the distance of web domains from initial web domains of a web crawl as a means to avoid low quality web sites was also studied. Results of this work and challenges for the future are summarised in Section 4.

2 Removing Spam from English Web Corpus Through Supervised Learning

This section describes training and evaluation of a supervised classifier to detect spam in web corpora.

[3] Source: `http://masterclasspolska.pl/forum/`, accessed in December 2015.

We have manually annotated a collection of 1630 web pages from various web sources from years 2006 to 2015.[4] To cover the main topics of spam texts observed in our previously built corpora, we included 107 spam pages promoting medication, financial services, commercial essay writing and other subjects.

Both phrase level and sentence level incoherent texts (mostly keyword insertions, n-grams of words stitched together or seemingly authentic sentences not conveying any connecting message) were represented. Another 39 spam documents coming from random web documents identified by annotators were included. There were 146 positive instances of spam documents altogether.

Table 1. Comparison of the 2015 English web corpus before and after spam removal using the classifier. Corpus sizes and relative frequencies (number of occurrences per million words) of selected words are shown. By reducing the corpus to 55 % of the former token count, phrases strongly indicating spam documents such as *cialis 20 mg*, *payday loan*, *essay writing* or *slot machine* were almost removed while innocent phrases not attracting spammers from the same domains such as *oral administration*, *interest rate*, *pass the exam* or *play games* were reduced proportionally to the whole corpus.

	Original corpus	**Clean corpus**	**Kept**
Document count	58,438,034	37,810,139	64.7 %
Token count	33,144,241,513	18,371,812,861	55.4 %
Phrase	**Original hits/M**	**Clean hits/M**	**Kept**
viagra	229.71	3.42	0.8 %
cialis 20 mg	2.74	0.02	0.4 %
aspirin	5.63	1.52	14.8 %
oral administration	0.26	0.23	48.8 %
loan	166.32	48.34	16.1 %
payday loan	24.19	1.09	2.5 %
cheap	295.31	64.30	12.1 %
interest rate	14.73	9.80	36.7 %
essay	348.89	33.95	5.4 %
essay writing	7.72	0.32	2.3 %
pass the exam	0.34	0.36	59.4 %
slot machine	3.50	0.99	15.8 %
playing cards	1.01	0.67	36.8 %
play games	3.55	3.68	53.9 %

The classifier was trained using FastText [5] and applied to a large English web corpus from 2015. The expected performance of the classifier was evaluated using a 30-fold cross-validation on the web page collection. Since our aim was to remove as much spam from the corpus as possible, regardless false positives,

[4] The collection is a part of another experiment co-authored by us.

the classifier top label probability threshold was set to prioritize recall over precision.

The achieved precision and recall were 71.5 % and 70.5 % respectively. Applying this classifier to an English web corpus from 2015 resulted in removing 35 % of corpus documents still leaving enough data for the corpus use.

An inspection of the cleaned corpus revealed the relative count of usual spam related keywords dropped significantly as expected while general words not necessarily associated with spam were affected less as can be seen in Table 1.

Table 2. Top collocate objects of verb 'buy' before and after spam removal. Corpus frequency of the verb: 14,267,996 (original), 2,699,951 (cleaned) – 81 % reduction by cleaning (i.e. more than the average reduction of a word in the corpus). The highest scoring lemmas are displayed. Frequency denotes the number of occurrences of the lemma as a collocate of the headword in the corpus. The score represents the typicality value (calculated by collocation metric LogDice [11] here) indicating how strong the collocation is.

Original lemma	frequency	score	Clean lemma	frequency	score
viagra	569,944	10.68	ticket	52,529	9.80
ciali	242,476	9.56	house	28,313	8.59
essay	212,077	9.17	product	37,126	8.49
paper	180,180	8.93	food	24,940	8.22
levitra	98,830	8.33	car	20,053	8.18
uk	93,491	8.22	book	27,088	8.09
ticket	85,994	8.08	property	17,210	7.88
product	105,263	8.00	land	15,857	7.83
cialis	71,359	7.85	share	12,083	7.67

To show the impact of the method on data used in real applications, Word Sketches of selected verb, nouns and adjectives in the original corpus and the cleaned corpus were compared. A Word Sketch is a table like report providing a collocation and grammatical summary of the word's behaviour that is essential for lexicography e.g. to derive the typical context and word senses of headwords in a dictionary. [6,1]. To create a good entry in a dictionary, one has to know strong collocates of the headword. We will show better collocates are provided by the cleaned corpus than the original version in the case of selected headwords.

Table 2 shows that top collocates of verb 'buy' in relation 'objects of verb' were improved a lot by applying the cleaning method to the corpus. It is true that e.g. 'buy viagra' or 'buy essay' are valid phrases, however looking at random concordance lines of these collocations, vast majority come from computer generated un-grammatical sentences.

Comparison of modifiers of noun 'house' in Table 3 reveals that the Word Sketch of a seemingly problem-free headword such as 'house' can be polluted by a false collocate – 'geisha'. Checking random concordance lines for co-occurrences of

'house' and 'geisha', almost none of them are natural English sentences. While 'geisha' is the fifth strongest collocate in the original corpus, it is not present among top 100 collocates in the cleaned version.

Table 3. Top collocate modifiers of noun 'house' before and after spam removal. Corpus frequency of the noun: 10,873,053 (original), 3,675,144 (cleaned) – 66 % reduction.

Original lemma	frequency	score	Clean lemma	frequency	score
white	280,842	10.58	publishing	20,314	8.63
opera	58,182	8.53	open	39,684	8.47
auction	41,438	8.05	guest	13,574	7.94
publishing	41,855	8.02	opera	9,847	7.67
geisha	38,331	7.95	old	32,855	7.64
open	37,627	7.78	haunted	9,013	7.58
old	73,454	7.52	auction	8,240	7.40
guest	28,655	7.44	manor	7,225	7.28
country	26,092	7.07	bedroom	7,717	7.26

The last comparison in Table 4 showing nouns modified by adjective 'green' is an example of cases not changed much by the cleaning.It is worthy of noting that apart from other words in this evaluation, the relative number of hits of adjective 'green' in the corpus was decreased less than the whole corpus. Although the classifier deliberately prefers recall over precision, the presence of non-spam words in the corpus was reduced less than the count of 'spam susceptible' words.

Table 4. Top collocate nouns modified by adjective 'green' before and after spam removal. Corpus frequency of the adjective: 2,626,241 (original), 1,585,328 (cleaned) – 40 % reduction (less than the average in the corpus).

Original lemma	frequency	score	Clean lemma	frequency	score
tea	86,031	10.04	tea	45,214	9.94
light	54,991	8.74	light	33,069	8.86
bean	28,724	8.63	space	51,830	8.72
egg	26,150	8.45	roof	17,916	8.72
space	55,412	8.19	bean	15,398	8.52
vegetable	20,906	8.16	economy	24,181	8.21
roof	18,910	8.1	energy	18,101	7.8
leave	16,712	7.74	infrastructure	13,331	7.69
economy	25,261	7.72	leave	9,754	7.69

3 Removing Spam from Estonian Web Corpus Through Semi-manual Classification of Web Sites

Unlike the spam classification of English web pages described in the previous chapter, where human annotators identified a small set of spam documents representing various non-text types, the annotators classified whole web domains this time. An Estonian web corpus crawled in 2019 was used in this experiment. Similarly to our previous result, supervised learning using FastText was employed to classify the corpus.

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. In this case, that work was done by Kristina Koppel from the Institute of Estonian Language at University of Tartu in several days.

Furthermore, it is efficient to check the most represented domains in the corpus. Thus a lot of non-text can be eliminated while obtaining a lot of training web pages at the same time. Spam documents coming from less represented web domains can be traced by the classifier once it is built.

A list of 1,000 Estonian web sites with the highest count of documents or the highest count of tokens in the corpus was used in the process of manual quality checking. There were also path prefixes covering at least 10 % of all paths within each site available to provide information about the structure of the domain. If the site was known to the human annotator, it was marked as good without further checks. If the site name looked suspiciously (e.g. a concatenation of unrelated words, mixed letters and numbers, or a foreign TLD), the annotator checked the site content on the web or its concordance lines in Sketch Engine.

Site name rules were formulated by observation of bad web domains. E.g. all hosts starting with `ee.`, `est.`, or `et.` under generic TLDs `.com`, `.net`, `.org`[5] were marked as non-text since there was machine translated content usually observed in these cases.

77 % of web pages in the corpus were semi-manually classified this way. 16 % of these documents were marked as computer generated non-text, mostly machine translated. 6 % of these documents were marked as bad for other reasons, generally poor quality content.

A binary classifier was trained using FastText on good and non-text web pages. URL of a page, plaintext word forms and 3 to 6 tuples of plaintext characters were the features supplied to FastText. 10 fold cross-validation was carried out to estimate the classifier's performance. Documents from the same web site were put in the same fold to make sure there was not the same content or the same URL prefix in multiple folds. Since the ratio of good to non-text samples in the data was approximately 77:16, the baseline accuracy (putting

[5] Excluding `et.wikipedia.org`.

all samples in the larger class) was 0.826. Despite the rather high baseline, the classifier performed well. FastText reported fold-averaged precision around 0.94 and recall from 0.93 to 0.76 based on the label probability threshold.

The final classifier was applied to the part of the corpus that had not been checked by the human annotator. 100 positive, i.e. non-text, and 100 negative, i.e. good, web pages were randomly selected for inspection. Kristina Koppel and Margit Langemets from the same institution checked the URL and plaintext[6] of each page. Three minimal probabilities of the top label were tested. The result precision and recall can be seen in Figure 1.

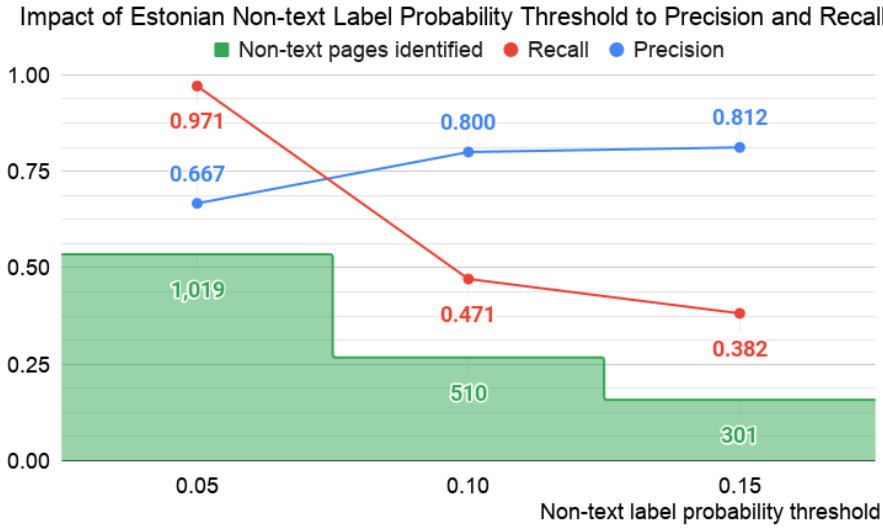


Fig. 1. Evaluation of the final binary spam classifier on documents not previously checked by a human annotator in Estonian web corpus. Precision and recall were estimated for minimal probabilities of the non-text label from 0.05 to 0.15. Since we aim for a high recall, the performance with the non-text label threshold set to 0.05 is satisfying. A higher threshold leads to an undesirable drop of recall.

It can be observed the recall dropped a lot with an increasing threshold. Therefore, the final top label probability applied to the corpus was set just to 0.05 to keep the recall high. We do not mind false positives as long as most of non-text is removed. We consider this setup and result as both time efficient and well performing. It will be applied to web corpora in other languages in cooperation with native speaker experts in the future.

Since web crawler SpiderLing [15], used to obtain the data, measures the distance of web domains from the initial domains, the value can be used to

[6] Texts were cropped to first 2,000 characters to speed up the process.

estimate the quality of the content of a web site. If our hypothesis was true, domains close to the seeds should be crawled more than domains far from the seeds.

To prove or reject the hypothesis, the classification of spam from the previous experiment was put into a relation with the domain distance of the respective good or bad documents. Both semi-manual and machine classification web pages were included. The binary classification of texts – good and bad labels – aggregated by the distance of the respective web sites from seed domains is displayed on Figure 2. The evaluation does not support the hypothesis much, at least in the case of the Estonian web.

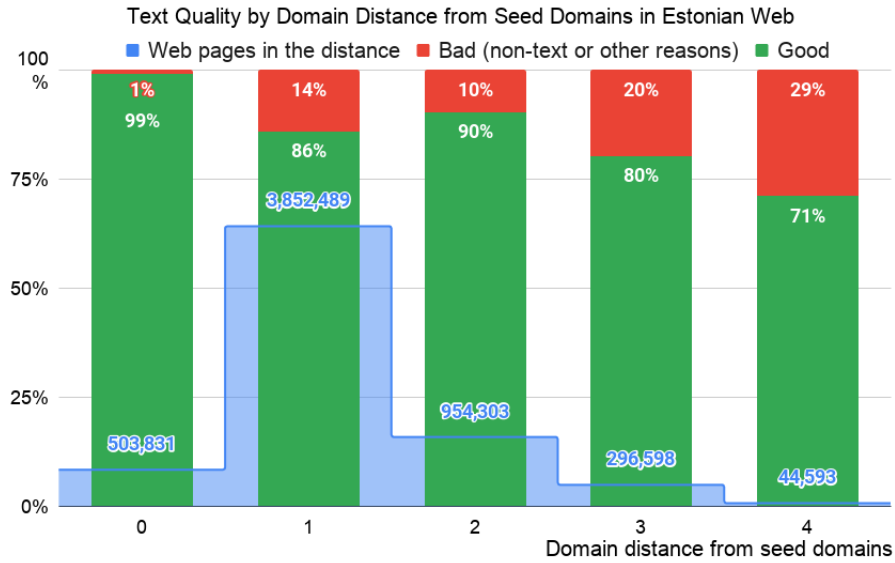


Fig. 2. Evaluation of the relation of the distance of web domain from the initial domains to the presence of non-text on the sites. Web pages of distances 0 to 4 classified semi-manually or by the spam classifier were taken into account. Two thirds of the pages were in distance 1. The percentage of good and bad documents within the same domain distance is shown. The presence of non-text in the data is notable from distance 1.

To sum up the findings of our experiments with Estonian web corpus:

1. A non-text classifier with a very high recall (at the cost of precision) can be trained on human annotated good and bad web sites.
2. The annotation process can be quite efficient: Checking web domains most represented in the corpus produces sufficient samples to classify the rest.
3. It is beneficial to start the crawling from trustworthy, quality content sites. However, there is non-text on web sites linked from the initial sites. The

domain distance is related to the presence of non-text but the correlation is not strong enough to make it an important feature in spam removal.

4 Conclusion and Future Challenges

Two experiments of spam removal based on supervised learning using FastText were presented in this paper.

A classifier trained on manually identified spam documents was applied to a recent English web corpus. The classifier was set to prefer recall at the cost of greatly reducing the size of the result corpus. Although the evaluation of the classifier on the training set reports a far from perfect recall of 71 %, it was able to notably decrease the presence of spam related words in the corpus.

An extrinsic evaluation was carried out by comparing the original data and the cleaned version in a lexicography oriented application: Relative corpus frequencies of words and Word Sketches of grammatical relations that could be used to make a dictionary entry for selected verb, noun and adjective were compared in the experiment.

Another experiment with a smaller Estonian corpus was carried out. An efficient human annotation lead to using more than two thirds of the corpus as training data for the spam classifier. The evaluation of the classifier shows a very high recall of 97 % was reached.

We understand the process can take more time for large internet languages such as English, Spanish, Russian or Chinese. We admit the number of sites in our Estonian experiment is small in comparison to these languages. Nevertheless we believe this is a good way to go for all languages. After all, Google needed human intervention to identify certain types of spam too.[7]

Although promising results were shown, we still consider computer generated non-text the main factor decreasing the quality of web corpora.

Computer generated text is on the rise. Although starting the crawl from a set of trustworthy seed domains, measuring domain distance from seed domains and not deviating too deep from the seed domains using hostname heuristics are ways to avoid spam, a lot of generated non-text will still be downloaded.

Machine translation is a specific subcase. Although there might exist a solution – watermarking the output of statistical machine translation – suggested by [16], we are not aware of the actual spread of this technique.

Strategies of non-text detection using language models will just compete with the same language models generating non-text. Nevertheless, the web will remain the largest source of text corpora.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

[7] Document 'Fighting Spam' accessed at `http://www.google.com/insidesearch/howsearchworks/fighting-spam.html` in January 2015.

References

1. Baisa, V., Blahuš, M., Cukr, M., Herman, O., Jakubíček, M., Kovář, V., Medveď, M., Měchura, M., Rychlý, P., Suchomel, V.: Automating dictionary production: a tagalog-english-korean dictionary from scratch. In: Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference. 1-3 October 2019, Sintra, Portugal. pp. 805–818 (2019)
2. Baisa, V., Suchomel, V.: Skell: Web interface for english language learning. In: Proceedings of 8th Workshop on Recent Advances in Slavonic Natural Languages Processing. pp. 63–70. Brno (2014)
3. Baroni, M., Bernardini, S.: Bootcat: Bootstrapping corpora and terms from the web. In: Proceedings of International Conference on Language Resources and Evaluation (2004)
4. Gyongyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: First international workshop on adversarial information retrieval on the web (AIRWeb 2005) (2005)
5. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
6. Kilgarriff, A., Baisa, V., Busta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The sketch engine: Ten years on. Lexicography **1**(1), 7–36 (2014)
7. Kilgarriff, A., Suchomel, V.: Web spam. In: Stefan Evert, Egon Stemle, P.R. (ed.) Proceedings of the 8th Web as Corpus Workshop (WAC-8) @Corpus Linguistics 2013. pp. 46–52 (2013)
8. Marek, M., Pecina, P., Spousta, M.: Web page cleaning with conditional random fields. In: Building and Exploring Web Corpora: Proceedings of the Fifth Web as Corpus Workshop, Incorporationg CleanEval (WAC3), Belgium. pp. 155–162 (2007)
9. Ntoulas, A., Najork, M., Manasse, M., Fetterly, D.: Detecting spam web pages through content analysis. In: Proceedings of the 15th international conference on World Wide Web. pp. 83–92. ACM (2006)
10. Pomikálek, J.: Removing boilerplate and duplicate content from web corpora. Ph.D. thesis, Masaryk University (2011)
11. Rychlý, P.: A lexicographer-friendly association score. In: Proceedings of 2nd Workshop on Recent Advances in Slavonic Natural Languages Processing. pp. 6–9 (2008)
12. Schäfer, R., Bildhauer, F.: Web Corpus Construction, vol. 6. Morgan & Claypool Publishers (2013)
13. Spoustová, J., Spousta, M.: A high-quality web corpus of czech. In: Proceedings of Eighth International Conference on Language Resources and Evaluation. pp. 311–315 (2012)
14. Suchomel, V.: Removing spam from web corpora through supervised learning using fasttext. In: Proceedings of the Workshop on Challenges in the Management of Large Corpora and Big Data and Natural Language Processing (CMLC-5+BigNLP) 2017 including the papers from the Web-as-Corpus (WAC-XI) guest section. pp. 56–60. Birmingham (2017)
15. Suchomel, V., Pomikálek, J.: Efficient web crawling for large text corpora. In: Adam Kilgarriff, S.S. (ed.) Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 39–43. Lyon (2012)
16. Venugopal, A., Uszkoreit, J., Talbot, D., Och, F.J., Ganitkevitch, J.: Watermarking the outputs of structured prediction with an application in statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1363–1372. Association for Computational Linguistics (2011)

17. Versley, Y., Panchenko, Y.: Not just bigger: Towards better-quality web corpora. In: Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 44–52 (2012)

Evaluating Russian Adj-Noun Word Sketches against Dictionaries: a Case Study

Maria Khokhlova

St Petersburg State University
Universitetskaya emb. 7-9-11
199034 St Petersburg, Russia
m.khokhlova@spbu.ru

Abstract. The paper discusses adj-noun word sketches produced for 20 Russian headwords. We analysed the differences between the output and collocations extracted from Russian dictionaries and also validated the collocates by expert evaluation. The aim was to study to what extent their data coincide with each other and to investigate how collocations presented in dictionaries are reflected in a large Web corpus. The comparison with the gold standard shows low precision whereas expert evaluation gives higher values. LogDice tend to extract more peculiar examples compared to joint frequency according to human assessment.

Keywords: Word sketches, Collocations, Evaluation, Dictionaries, Russian language.

1 Introduction

Lexicography and corpus linguistics become more user-oriented. Sketch Engine was one of the first systems to facilitate research by taking over most of the routine procedures [13]. The word sketches greatly influenced applied linguistics helping to represent collocational behaviour in the form of convenient tables, which until then had to be found by separate queries or with other filters. More than 20 years have passed since the appearance of the first word sketch grammar. Therefore, we see the need to re-evaluate them, analyze possible issues, and also outline ways to solve them. We currently work on creating a gold standard of collocability for the Russian language [6], which can be used as a source of reference data. We consider collocations extracted in Russian dictionaries and analyse how they are reflected in a corpus of contemporary Russian and hence are presented as word sketches. Thus, the purpose of our research is to compare word sketches with verified lexicographic data, i.e. trace the intersection between data collected by experts and automatically extracted from an up-to-date corpus.

The paper is structured as follows. The Introduction presents the basic idea of the research. The next section provides a brief overview of the related work.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020, pp. 125–131, 2020.

Section 3 discusses the methods and relevant notions, i.e. the word sketch rules and gold standard used during the analysis. The next section examines the results of the experiment while the last one concludes the paper and offers future perspectives.

2 Background

A profound evaluation of word sketches was presented by A. Kilgarriff et al. [8] for four languages (Dutch, English, Japanese and Slovene). The authors differentiate between developer and user approach concentrating on the latter. The article [5] introduces Russian word sketches and describes Russian sketch grammar. The Russian language can be seen as underestimated in various kinds of analysis therefore the evaluation of the output of Russian word sketches may yield nontrivial results.

3 Methods

In the paper we also adhere to 'user evaluation' perspective [8] and will address to the dictionaries as sources of expert data.

3.1 Gold Standard

In contrast to the approach presented in [8] we use dictionaries as verified sources, i.e. we consider them as containing data that has already been approved by experts. As mentioned above, during our work on the gold standard of Russian collocability [6,7], we collected examples from six different Russian dictionaries. During the study we scrutinized the following ones:

1. two explanatory dictionaries, i.e. the Dictionary of the Russian Language [2]; the Large Explanatory Dictionary of the Russian Language [10];
2. three collocation dictionaries [1,11,12];
3. an online dictionary [9].

Within the analysis we considered attributive collocations built according to the "adjective / participle + noun" model. At the moment, the database includes more than 15 thousand units of such a type. The dictionary data from the gold standard is suitable for evaluating the precision of word sketches output. Since word sketches are shown for a headword, we decided to consider a number of nouns that form this type of collocations.

We analysed the most frequent headwords presented in collocations from the gold standard, namely the ones having a variety of syntagmatic relations. The 20 selected headwords turn to highly productive, i.e. form a wide range of collocations (the precise number is given in parentheses): *sila* 'force' (97), *uspekh* 'success' (59), *bor'ba* 'fight' (55), *toska* 'boredom' (54), *lyubov'* 'love' (49), *interes* 'interest' (46), *delo* 'case' (43), *bolezn'* 'illness' (42), *radost'* 'joy' (40), *pamyat'*

'memory' (40), *krasota* 'beauty' (38), *znacheniye* 'meaning' (37), *chuvstvo* 'sense' (36), *sistema* 'system' (36), *nenavist'* 'hate' (36), *um* 'intellect' (35), *strast'* 'passion' (34), *rol'* 'role' (34), *kholod* 'cold' (33), *usiliye* 'effort' (32). As one can see the majority of the nouns refer to emotions and abstract notions.

3.2 Experiment: Settings

As authors in [8] rightly note, it is also necessary to evaluate those collocations that could be seen as potential candidates for the inclusion in the dictionary, so we will evaluate recall in two ways: 1) by comparing to the dictionary data; 2) by an expert's assessment.

We confine ourselves to the first 50 examples produced by word sketches. From a user's perspective it is reasonable to process brief lists of collocates. Sketch Engine enables ranging word sketches according to two measures (namely, logDice and joint frequency). Since we analyse top-50 collocations, the results can differ; hence we decided to use both types of output in our evaluation. In other words it can seen as an evaluation of not only word sketches but also of two measures. Following the approach by S. Evert [3] we compute precision as proportion of collocations (identified either in the gold standard or by expert evaluation) in the output and recall as proportion of collocations from the gold standard that were correctly extracted from the corpus.

RuTenTen corpus is one of the largest Russian corpora [4], so we chose it for our experiment and expect to see the widest range of collocations extracted from it.

3.3 Word Sketch Grammar

Collocations based on the "adjective / participle + noun" model will be in the focus of our attention, e.g., *prakticheskoye znacheniye* 'practical meaning', *zhiznennyy uspekh* 'life success', *oslepitel'naya krasota* 'dazzling beauty', etc. In [2] we described the rules for the Russian language, which were implemented for generating word sketches. Below one can see a subset from the so called "word sketch grammar" which takes into account this type of collocations.

```
*DUAL
=amodifier/modifies
2:adj 1:noun & agree(1,2)
2:adj 3:adj 1:noun & agree(1,2) & agree(1,3)
2:adj 3:adj 4:adj 1:noun & agree(1,2) & agree(1,3) & agree(1,4)
2:adj 3:adj 4:adj 5:adj 1:noun & agree(1,2) & agree(1,3) & agree(1,4)
  & agree(1,5)
2:adj [word="  "|word=" "] 3:adj 1:noun & agree(1,2) & agree(1,3)
2:adj [word=","]? 4:adj [word="  "|word=" "] 3:adj 1:noun & agree(1,2)
  & agree(1,3) & agree(1,4)
2:adj [word=","]? 4:adj [word=","]? 5:adj [word="  "|word=" "] 3:adj
  1:noun & agree(1,2) & agree(1,3) & agree(1,4) & agree(1,5)
2:adj [word=","] 3:adj 1:noun & agree(1,2) & agree(1,3)
2:adj [word=","] 3:adj [word=","] 4:adj 1:noun & agree(1,2) & agree(1,3)
  & agree(1,4)
```

```
2:adj [word=","] 3:adj [word=","] 4:adj [word=","] 5:adj 1:noun
  & agree(1,2) & agree(1,3) & agree(1,4) & agree(1,5)
```

The Russian language is characterized by rich morphology and has a high number of inflections, therefore, an adjective or a participle must have the same gender and case with the noun to which they belong (in the above given rules this agreement is marked by 'agree' showing in parentheses the numbers of words involved in this relation). The above mentioned word sketch rules cover noun phrases and can be illustrated by the following examples:

1. adj-noun (e.g. *temperaturnyy rezhim* 'temperature regime');
2. adj-adj-noun (e.g. *vysokochastotnyy elektricheskiy tok* 'high-frequency electrical current');
3. adj-adj-adj-noun (e.g., *global'naya sputnikovaya navigatsionnaya sistema* 'global navigation satellite system');
4. adj-adj-adj-adj-noun (e.g., *kitayskiy zelenyy baykhovyy krupnolistovoy chay* 'Chinese green loose large leaf tea');
5. adj-conj-adj-noun (e.g. *mobil'nyy ili domashniy telefon* 'mobile or home phone number');
6. adj-,-adj-conj-adj-noun (e.g., *tekhnicheskaya, informatsionnaya i reklamnaya podderzhka* 'technical, information and advertising support');
7. adj-,-adj-,-adj-conj-adj-noun (e.g., *administrativnoye, pensionnoye, sotsial'noye i trudovoye zakonodatel'stvo* 'administrative, pension, social and labour law');
8. adj-,-adj-noun (e.g. *federal'nyy, regional'nyy uroven'* 'federal, regional level');
9. adj-,-adj-,-adj-noun (e.g., *neftyanaya, khimicheskaya, pischevaya promyshlennost'* 'oil, chemical, food industry');
10. adj-,-adj-,-adj-,-adj-noun (e.g., *doshkol'noye, obscheye, dopolnitel'noye, vyssheye obrazovaniye* 'preschool, general, supplementary, higher education').

These ten cases describe collocations of varying length taking into account a certain distance between nodes and collocates. Adjectives can be separated by commas or combined by conjunctions *i* 'and' and *ili* 'or'.

4 Results

The output showed collocates produced with morphological errors that can be accounted for several reasons. The results list token collocates belonging to the same lemmata but representing different cases, numbers or genders. For example, word sketches list *vol'nyy* 'free' (masculine gender) and *vol'naya* 'free' (feminine gender) as collocates for the lemma *bor'ba* 'fight'. This type of errors leads to the discrepancies in frequencies and hence to the false output and false ranging by both statistical measures.

For adj-noun collocations, representation of participles as verb forms can be seen as a certain problem. For example, one can find the following collocates for the headword *krasota* 'beauty': *zavorazhivat'* 'to bewitch' instead of *zavorazhivayuschiy* 'bewitching', *potrysat'* 'to amaze' instead of *potrysayuschiy* 'stunning'. However there are word sketches listing both verbs and participles as frequent collocates (e.g. *dominirovat'* 'to dominate' and *dominiruyuschiy* 'dominant'

for the headword *rol'* 'role'). It could be more convenient for users (especially for Russian language learners) to see forms of the participle in the apropriate word sketch table (amodifier/modifies), i.e. *zavorazhivayuschiy* or *potrysayuschiy*.

Most of the errors in lemmatisation was found for the collocates with the headword *bolezn'* 'disease'. This can be due to the fact that they represent themselves special terms and therefore are absent in the morphological dictionary. Also a large number of incorrect results was produced for the headword *usiliye* 'effort'. The output shows verbs (instead of participles) and incorrect gender and case forms for collocates.

The total number of such errors equals to 7.4% for logDice and 4% for joint frequency respectively.

LogDice tend to extract more peculiar collocations. For example, among the first 50 results we found *tsepkaya pamyat'* 'tenacious memory' and *fotograficheskaya pamyat'* 'photographic memory', i.e. these collocations can be listed in entries of dictionaries for Russian language learners. The joint frequency measure yield yet less promising collocations among the top 50 ones.

Table 1 shows the results for the precision and recall computed when compared with the gold standard and expert assessment. The least number of the examples were found for the headword *pamyat'* 'memory'. This can be due to the fact that the corpus contains contemporary texts showing mostly occurrences for this noun with the meaning "computer memory" while dictionaries list examples for other meanings.

Table 1. Precision and recall.

Headword		Precision (dictionary, logDice)	Precision (expert, logDice)	Precision (dictionary, freq)	Precision (expert, freq)	Recall (dictionary, logDice)	Recall (dictionary, freq)
bolezn'	'illness'	0.48	0.88	0.52	0.84	0.57	0.62
bor'ba	'fight'	0.38	0.54	0.42	0.54	0.35	0.38
chuvstvo	'sense'	0.16	0.24	0.28	0.30	0.22	0.39
delo	'case'	0.18	0.36	0.22	0.36	0.21	0.26
interes	'interest'	0.28	0.46	0.24	0.36	0.30	0.26
kholod	'cold'	0.42	0.50	0.36	0.42	0.64	0.55
krasota	'beauty'	0.36	0.56	0.36	0.44	0.47	0.47
lyubov'	'love'	0.32	0.66	0.30	0.56	0.33	0.30
nenavist'	'hate'	0.32	0.38	0.38	0.44	0.44	0.53
pamyat'	'memory'	0.20	0.58	0.18	0.42	0.25	0.23
radost'	'joy'	0.38	0.46	0.36	0.38	0.48	0.45
rol'	'role'	0.48	0.56	0.44	0.50	0.71	0.65
sila	'force'	0.46	0.84	0.44	0.70	0.24	0.23
sistema	'system'	0.08	0.40	0.08	0.36	0.11	0.11
strast'	'passion'	0.32	0.42	0.36	0.42	0.47	0.53
toska	'boredom'	0.40	0.44	0.50	0.54	0.37	0.46
um	'intellect'	0.38	0.52	0.34	0.38	0.51	0.49
usiliye	'effort'	0.18	0.34	0.24	0.28	0.28	0.38
uspekh	'success'	0.56	0.64	0.40	0.52	0.47	0.34
znacheniye	'meaning'	0.28	0.40	0.36	0.02	0.38	0.49

The mean precision computed against gold standard was quite low and equal to 0.33 and 0.34 for logDice and joint frequency respectively. The expert analysis revealed fascinating collocates among word sketches and hence raised the mean precision to 0.51 and 0.44 respectively. LogDice showed again more interesting results according to human assessment compared to joint frequency (e.g. *kriticheskoye znacheniye* 'critical value' or *shkurnyy interes* 'selfish interest').

5 Conclusion

We examined word sketches for 20 nouns that form the largest number of collocations according to six Russian dictionaries. Thus, this formal evaluation was based on a comparison between corpus and lexicographic data. In total 1,000 word sketches per measure (logDice and joint frequency) were analyzed. The analysis showed that the precision of the word sketches output is a bit low with regard to the data extracted from different Russian dictionaries while they show higher and more promising results assessed by expert evaluation. At least half of the produced word sketches can be called "true collocations" and can be included into dictionaries (that do not list them yet) and here we can foresee broad perspectives.

Although logDice measure shows quite similar quantitative results with joint frequency, however, it turns out to be much more successful for extracting and ranking word sketches according to the expert assessment. This confirms the choice of this measure as the default one in Sketch Engine. These results can be also relevant for further evaluation of statistical measures used for collocation extraction. In future we plan to evaluate other models described in the word sketch grammar and analyse more headwords.

Acknowledgments. This work was supported by the grant of the Russian Science Foundation (Project No. 19-78-00091).

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. 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).
3. Evert, S.: Corpora and collocations. In: Corpus Linguistics. An International Handbook 2. pp. 1212-–1248. (2008)
4. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. In: Proceedings of the 7th International Corpus Linguistics Conference CL 2013, the United Kingdom, July 2013, pp. 125–127 (2013).

5. Khokhlova, M.: Applying Word Sketches to Russian. In: Proceedings of Raslan 2009. Recent Advances in Slavonic Natural Language Processing, pp. 91–99. Masaryk University: Brno (2009)
6. 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)
7. 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)
8. Kilgarriff, A., Kovar, V., Krek, S., Srdanovic, I., and Tiberius, C.: A quantitative evaluation of word sketches. In: Proceedings of the XIV Euralex International Congress, pp. 372–379. Fryske Academy: Leeuwarden (2010)
9. 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 2020.
10. The Large Explanatory Dictionary of the Russian Language [Bol'shoy tolkovyy slovar' russkogo yazyka]. S.A. Kuznetsov (ed.). Norint: St. Petersburg (1998).
11. Oubine, I.: Dictionary of Russian and English Lexical Intensifiers [Slovar' usilitel'nykh slovoso-chetaniy russkogo I angliyskogo yazykov]. Russian Language: Moscow (1987).
12. 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).
13. Sketch Engine Homepage, `http://www.sketchengine.eu`. Last accessed 14 Nov 2020

Subject Index

collocations 125
corpus 95, 113
corpus building 95, 113
corpus statistics 103
Czech 79

data mining 13
dataset management 23
date recognition 67

electronic health records 13
English 113
Estonian 113
evaluation 37, 125

fastText 37, 55
formal concept analysis 47
French 95

grammar checker 79

historical texts 3

information retrieval 37

language identification 87
language modeling 37
lyrics 95

machine learning 23, 55
multilingual 67

N-grams 87
named entity recognition 13
neology 95

optical character recognition 3
optimization 23

parsing 103
Polish 13
prepositional constructions 103

question answering 23

R'lyehian 87
rap music 95
reproducibility 55
Russian 103, 125

semantic classes 103
Slavic languages 13
substandard language 95
supervised learning 113

temporal expression 67
text classification 37
text processing 95
Transparent Intensional Logic 47

VerbaLex 79

web corpora 113
web spam 113
word analogy 55
word sketches 125
word vectors 37, 55
word2vec 37, 55

zeugma 79

Author Index

Albert, A. 47
Anetta, K. 13

Boyarsky, K. 103

Golovina, A. 103

Horák, A. 23

Khokhlova, M. 125
Kozlova, A. 103

Lupták, D. 37

Medková, H. 79
Medveď, M. 23
Menšík, M. 47

Nevěřilová, Z. 67
Novotný, V. 3, 37, 55, 87

Patschka, V. 47
Podhorná-Polická, A. 95

Sabol, R. 23
Sojka, P. 37
Stará, M. 87
Starý, M. 67
Suchomel, V. 113

Valčík, J. 67

Zakharov, V. 103

Štefánik, M. 37

RASLAN 2020

Fourteenth 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 2020

ISBN 978-80-263-1600-8
ISSN 2336-4289