

RASLAN 2023
Recent Advances in Slavonic
Natural Language Processing

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

RASLAN 2023

**Recent Advances in Slavonic Natural
Language Processing**

**Seventeenth Workshop on Recent Advances
in Slavonic Natural Language Processing,
RASLAN 2023**

**Kouty nad Desnou, Czech Republic,
December 8–10, 2023
Proceedings**

**Tribun EU
2023**

Proceedings Editors

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

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

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

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

Editors © Aleš Horák, 2023; Pavel Rychlý, 2023; Adam Rambousek, 2023

Typography © Adam Rambousek, 2023

Cover © Petr Sojka, 2010

This edition © Tribun EU, Brno, 2023

ISBN 978-80-263-1793-7

ISSN 2336-4289

Preface

This volume contains the Proceedings of the Seventeenth Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2023) , organized by the NLP Consulting, s.r.o. and held on December 8th–10th 2023 in Kouty nad Desnou, Jeseníky, Czech Republic.

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

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

- * text corpora and tagging
- * neural language modelling
- * 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 2023 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 17 papers were accepted, contributed altogether by 29 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 Barbora Stenglová. The \LaTeX expertise 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.

Table of Contents

I NLP Applications

Web-Based Annotation Tool for Instant Messaging Conversations	3
<i>Jaromír Plhák, Michaela Lebedíková, Michał Tkaczyk, and David Šmahel</i>	
Five Years of Language Services	11
<i>Zuzana Nevěřilová</i>	
CIVQA: Czech Invoice Visual Question Answering	23
<i>Šárka Ščavnická, Michal Štefánik, and Petr Sojka</i>	
Can We Detect ChatGPT-generated Texts in Czech and Slovak Languages?	35
<i>Petr Šigut and Tomáš Foltýnek</i>	

II Evaluation Methods

Does Size Matter?	47
<i>Michaela Denisová and Pavel Rychlý</i>	
Reproducibility and Robustness of Authorship Identification Approaches	57
<i>Adam Karásek and Zuzana Nevěřilová</i>	
Augmenting Stylometric Features to Improve Detection of Propaganda and Manipulation	67
<i>Radoslav Sabol and Aleš Horák</i>	
Predicting Style-Dependent Collocations in Russian Text Corpora	79
<i>Lada Petrushenko and Olga Mitrofanova</i>	

III Text Corpora

Semi-automatic Dictionary Creation for Czech	93
<i>František Kovařík</i>	
Development of the NVH Schema Format for Lexicographic Purposes	101
<i>Marek Medveď, Miloš Jakubíček, Vojtěch Kovář, and Tomáš Svoboda</i>	
Data Gathered with Automatic Tools from European Parliamentary Chambers	107
<i>Ota Mikušek</i>	

Towards Perfection of Machine Learning of Competing Patterns: The Use Case of Czechoslovak Patterns Development	113
<i>Ondřej Sojka and Petr Sojka</i>	

Verb-Object Collocations in the Russian Collocations Database: Linguistic and Statistical Properties	121
<i>Maria Khokhlova</i>	

IV Semantics and Language Modelling

Towards Using Speech Melody to Guide Large Language Models	133
<i>David Porteš</i>	

Fine-Grained Language Relatedness for Zero-Shot Silesian-English Translation	145
<i>Edoardo Signoroni</i>	

Creating an Annotated Health Record Dataset in a Limited-Resource Environment	157
<i>Kristof Anetta</i>	

Thematic Markers and Keywords on the Example of German Political Discourse	165
<i>Maria Khokhlova and Mikhail Koryshev</i>	

Subject Index	175
----------------------------	-----

Author Index	177
---------------------------	-----

Part I

NLP Applications

Web-Based Annotation Tool for Instant Messaging Conversations

Jaromír Plhák , Michaela Lebedíková , Michał Tkaczyk , and David Šmahel 

Faculty of Informatics, Masaryk University, Brno, Czech Republic
{xplhak, x450458, x245062, davs}@fi.muni.cz

Abstract. This paper presents a customized web-based annotation tool that allows users to annotate utterances in data from instant messaging applications. Efficient annotations are provided by a well-arranged user interface, operating using key-pressing and integration of an interactive annotation manual. Moreover, the interface for supervisors allows them to determine which utterances belong to the gold standard. We also provide information on two accomplished annotation tasks: annotating online risk phenomena with sparse occurrence (0.85% to 1.98%) and annotating social support that can be used to generate efficient detection models.

Keywords: Text annotation tool, online risky behavior, social support.

1 Introduction

Annotation tools are beneficial when developers create deep-learning NLP models, as they require a lot of high-quality data to make accurate predictions. Labeling of this training data is usually provided by human users who have good expertise in the target domain. This task is frequently very effort-intensive and time-consuming. Therefore, annotation tools should allow users to simplify the annotation process to improve their productivity and ensure data coherence and inter-annotator agreement (IAA).

Many text annotation tools are available for various text annotation tasks [1,3,4,5,8,9,10,11,17]. They allow the users to annotate words, sentences, and other text parts using specified tags and also label their relations and dependencies. Such tools also usually provide work distribution among team members and different user roles like annotators or supervisors. They also offer various levels of security, including role-based access, zero data sharing, or multi-factor authentication. Moreover, they provide functionalities like integration with external resources, annotation comparison, IAA calculation, or AI-assisted annotation.

However, as each annotation tool is developed for a given task or set of tasks, it is hard to use them for processing specific types of data. Within the project Modeling the future: Understanding the impact of technology on adolescent's well-being (FUTURE) [2], we acquired data from adolescents' Messenger and

WhatsApp conversations to generate efficient online risk and social support detection models [12,14,15]. This task presented several challenges that we needed to overcome: (1) we needed to anonymize the data, (2) we needed to parse exported data to suitable units of text that would be large enough to convey the meaning of the conversation and its topic, (3) the task of annotating complete utterances representing one user prompt. Taken together, the current solutions provided by other annotation tools have proven ineffective, time-consuming, and unable to meet our demands due to their complex GUI and general functionality.

Therefore, we designed and implemented a new web-based tool for annotating instant messaging (IM) conversations called **IRTIS Annotation Tool (IRTIS AT)**. Unlike most existing solutions, our tool allows for annotating complete utterances representing one user prompt, regardless of whether this prompt is one word or multiple sentences. Also, our solution relies on using key-pressing instead of a computer mouse to fasten the annotation process.

User: [REDACTED]

Please select your assignment:

☒ Active ☒ Unfinished ☒ Finished

Assignment	Burst	TagSet	Status	Conversations	Start time	Finish time
0	0_1_unique	0	Finished	3808/3808	18/01/2022, 14:04:27	04/07/2023, 17:57:54
1	0_1_unique_q_mark	0	Finished	578/578	18/01/2022, 08:37:45	12/07/2023, 20:37:44
2	0_1_same_q_mark	0	Finished	316/318	28/01/2022, 10:01:58	28/01/2022, 12:31:20
3	0_1_same_two_q_marks	0	Finished	31/31	28/01/2022, 09:50:58	28/01/2022, 10:01:41
4	0_2_unique	0	Unfinished	35/4065	13/07/2023, 12:48:28	NaN
5	0_2_unique_q_mark	0	Active	0/549	NaN	NaN

Fig. 1: Interface for annotation burst selection

2 IRTIS AT

The IRTIS AT allows users to annotate texts from online communicators like Messenger or WhatsApp. It processes files that can be manually exported from these apps via their export functionality [6,7]. During import to IRTIS AT, data are anonymized based on the algorithm presented in [16]. Subsequently, it

provides users with two basic interfaces: for annotators and for supervisors, where they check annotation disagreements. Annotator's interface allows the users to annotate utterances using specific tags from the tagset (predefined set of applicable tags). The example of the interface for annotation burst selection can be seen in Figure 1 and the interface for supervisor's annotations revision in Figure 2.

Potential users of the IRTIS AT are researchers who want to annotate data from online messaging applications in a well-arranged and intuitive way. It supports the annotators and supervisors, who can examine the amount of finished work and time spent on annotations.

Conversation: 2024 - 69 - 210 (36/4065)

0: NO TAG (0) 1: Informační podpora (1) 2: Emocionální podpora (2) 3: Zařazení do skupiny (5) 4: Uznání (6) 5: Nabídka pomoci (5) 21: Cizí jazyk (0)

#	Line	Ann 1	Ann 2	Tag
9	09.10.2019 19:04:47 - Kateřina Lendrová: Už měli laborky			
10	09.10.2019 19:05:39 - Pavel Ashby: Říkala písemku na příklady	1	1	
11	09.10.2019 19:06:23 - Darja Poljansky <OWNER>: Hej podle mě fakt ne ale jak chcete	1	1	
12	09.10.2019 19:06:37 - Pavel Ashby: tak ono je to easy	2		2
13	09.10.2019 19:22:07 - Darja Poljansky <OWNER>: Hej ten případ se sanema			
14	09.10.2019 19:22:13 - Darja Poljansky <OWNER>: U té práce			
15	09.10.2019 19:22:19 - Darja Poljansky <OWNER>: To kdy nastává			
16	09.10.2019 19:22:50 - Darja Poljansky <OWNER>: ?			
17	09.10.2019 19:24:32 - Pavel Ashby: když jedas na sanich?			
18	09.10.2019 19:24:46 - Darja Poljansky <OWNER>: No ale jaký pohyb			
19	09.10.2019 19:24:51 - Darja Poljansky <OWNER>: Škmo dolů?			
20	09.10.2019 19:25:23 - Pavel Ashby: wtf!!!			
21	09.10.2019 19:26:41 - Darja Poljansky <OWNER>: Se uklidni			
22	09.10.2019 19:26:49 - Darja Poljansky <OWNER>: Jen se snažím něco zjistit			
23	09.10.2019 19:26:58 - Pavel Ashby: vsak jaa jsem v klidu			
24	09.10.2019 19:27:04 - Pavel Ashby: jen nechapu jaký pohyb			
25	09.10.2019 19:27:07 - Pavel Ashby: dolu			

Previous Reset Following Go back Stop Annotation Finish burst Save this conversation and load next one

50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500
1550 1600 1650 1700 1750 1800 1850 1900 1950 2000 2050 2100 2150 2200 2250 2300 2350 2400 2450 2500 2550 2600 2650 2700 2750 2800 2850 2900 2950 3000
3050 3100 3150 3200 3250 3300 3350 3400 3450 3500 3550 3600 3650 3700 3750 3800 3850 3900 3950 4000 4050

Enlarge Hide

**ANOTAČNÍ MANUÁL:
PODPŮRNÉ INTERAKCE**

Verze 2. 23. 11, zkrácená verze

CO JE TO PODPORA (PODPŮRNÉ INTERAKCE ONLINE)?

Podpůrné interakce online mezi vrstevníky, my se zaměříme konkrétně na poskytování podpory ve smyslu komunikace, o které můžeme předpokládat, že u příjemce vyvolá pocit/přesvědčení, že někomu na něm záleží, je milován, vážen a oceňován, že někdo má o něj starost, je pro něj oporou, že mu někdo poskytne radu nebo užitečnou informaci, bude-li potřeba, pomůže mu a také, že je součástí skupiny, s jejíž členy tráví čas a má společné plány.

Pro každý typ podpory platí, že pokud výpověď v daném řádku nemá tento účel, neměla by být označena jako podpora.

1. Informační podpora

Je obsahem řádku něco z následujícího:

[dávání rad/tipů] [učení] [zpětné vazby] [předávání znalostí/informací] [situace, které druhý potřebuje a jsou mu nápomocné v řešení problému/pochopení situace, v které se ocitl]

• Z kontextu konverzace musí být zřejmé, že informace jsou pro příjemce PODPORU v nějakém konkrétním jednání/situaci [jsou nápomocné v řešení problému]

Fig. 2: Interface for supervisor's annotations revision

2.1 Functionality and Development Process

Initially, the following functionality for the IRTIS AT was discussed with supervisors and fine-tuned after testing the prototype:

- Data can be uploaded to the server, anonymized, and divided into conversations spaced by periods of non-communication longer than 60 minutes (currently done outside the tool).
- Tagsets can be specified and include selected tags (currently done manually using .json configuration files).
- Conversations are grouped in a series of messages called bursts. Bursts contain a limited number of conversations to prevent annotator fatigue (currently done outside the tool).
- Annotators have accounts and can be assigned to selected bursts (currently done manually using a .json configuration file).

- Annotators can choose the burst (from the assigned bursts) they want to work on.
- Conversations can be annotated by annotators using tags from tagset:
 - Tags can be selected using key pressing.
 - Importantly, annotators can select up to three additional tags, if applicable. For example, adding a T+ symbol means that the annotator assumes another online risk or social support category can be used for a given utterance. Moreover, as our annotation task was very complex, annotators sometimes were unsure whether a tag should be applied, especially in the training phase. Our tool allows them to express this uncertainty by using a question mark, and these can be later viewed by the annotation supervisors and fine-tuned.
 - Previous and subsequent conversations can be loaded on demand to assess the context of the conversations.
 - Annotation manual is integrated into GUI and allows annotators to access requested parts interactively.
- Supervisor mode allows supervisors to decide ambiguous cases where the annotators disagree with the tag.
- Annotated data can be exported into MS Excel sheet or .csv files for further processing and IAA calculation.
- Statistics about work progress will be displayed in the GUI (number of finished conversations, total number of conversations in the burst, tagset, starting and finishing times).
- Time spent on the annotation is logged for each annotator and burst.

Statistics about annotations

Show entries Search:

User	Hash	Real data	Burst	TagSet	Status	Start time	Finish time	Lines	Work time	Fully annotated	
Jana	██████████	False	0	0	Unfinished	03/12/2020, 11:19:34	NaN	449/4142	2:04:56	18/259	Details
Jarek	██████████	False	0	0	Unfinished	04/12/2020, 16:18:39	NaN	246/4142	1:21:20	15/259	Details
Jarek	██████████	False	47	1	Unfinished	14/03/2021, 22:11:46	NaN	0/10088	0:06:10	0/182	Details
Karolina	██████████	True	10	0	Finished	21/12/2020, 09:48:49	29/12/2020, 11:28:06	8003/8003	8:59:24	391/391	Details
Karolina	██████████	True	11	0	Finished	29/12/2020, 11:48:14	02/01/2021, 18:01:23	8008/8008	9:53:19	447/447	Details
Karolina	██████████	True	12	0	Finished	04/01/2021, 16:09:57	05/01/2021, 21:26:23	8145/8145	9:09:17	437/437	Details
Karolina	██████████	False	5	0	Finished	09/01/2021, 13:22:55	NaN	4004/4004	0:01:44	243/243	Details
Karolina	██████████	True	13	0	Finished	07/01/2021, 11:35:50	09/01/2021, 13:23:58	8148/8148	8:37:30	425/425	Details

Fig. 3: Annotation statistics for supervisors

IRTIS AT was implemented using Python/flask technology and deployed on the server within Masaryk University. The source code is available in the Gitlab repository [13].

xlsx generation

Select burst

ID: 16 Conversations: 3025-2-77, 3030-21-13, ...

Select tagset

ID: 1 Tag: Agrese, obtěžování, nenávistné proj., Problémy s duševní zdravím, Alkohol a drogy, ...

Select users

☐ Pavel (Finished)

☒ Michal (Finished)

☐ Miroslav (Finished)

☒ Generate tags with question mark?

Generate

Fig. 4: Interface for data export to MS Excel

2.2 Annotation Process

Before the IRTIS AT's implementation, annotation manuals were developed for **Online risky behavior** (aggression, harassment, hate; mental health; use of alcohol and drugs; sexual content and sexting), and provision of **Social support** (informational support; emotional support; social companionship; appraisal; and instrumental support). For each task, two annotators were trained for two months, and the manual was gradually refined based on their and supervisors' feedback. Finally, the annotators started to code randomly generated bursts of data. The occurrence of the category online risky behavior in our corpus was sparse (see Table 1). Therefore, as we need as many positive examples as possible, we developed a preliminary classifier to identify conversations with a higher chance of containing utterances. It was then used to generate bursts for annotation of this specific category.

In the next step, the gold standard was generated. A dedicated supervisor interface was designed and developed as one of the tool components. Using this, supervisors could solve disagreements between annotators and utterances with ambiguous tagging (see Figure 2).

The annotation component of the tool comprises two windows. The initial window offers an overview of batches that were assigned to the annotator. The following information is provided to each batch: ID number, name, ID of specific tagset, status, number of annotated units/number of all units in batch, starting time, and finish time (see Figure 1). After selecting a batch, the window for annotation opens (see Figure 2). Information about the annotated unit is displayed in the upper part of the annotation window. Below this information, virtual keys for each tag are displayed. Each key contains the name of the tag and key bindings. The annotation manual is displayed on the right side of the screen; it can be scrolled down or enlarged. The annotation user interface is displayed in the center part of the window. It comprises lines to be annotated and tags assigned to each line by the annotator. In the bottom part of the window, virtual functional keys are located. Those keys enable moving backward and forward across larger units (conversations), enlarging the number of rows displayed in the annotation user interface if a broader context is needed to decide, finish, and save the annotation.

The annotation process resulted in 272,465 utterances with online risky behavior tagset (ORB) and 196,772 with social support tagset (SocS). To detect the difficulty of annotating each tagset, we compare the time spent by every

Table 1: Overview of the number of annotated utterances and IAA (Cohen’s κ)

Category	Tagset	Annotated by at least one annotator	κ
Aggression, harassment, hate	ORB	5393 (1.98%)	.470
Mental health problems	ORB	3101 (1.14%)	.460
Alcohol, drugs	ORB	2301 (0.85%)	.609
Sexual content	ORB	3550 (1.30%)	.485
Informational support	SocS	9967 (5.07%)	.685
Emotional support	SocS	9669 (4.92%)	.639
Social companionship	SocS	3331 (1.70%)	.604
Appraisal	SocS	2524 (1.28%)	.650
Instrumental support	SocS	5317 (2.70%)	.599

annotator. Table 2 shows that the annotation time differs between two annotators by approximately 15%, and annotating social support was approximately three times slower than annotating online risky behavior. This is based on a higher density of social support categories in corpora, and, therefore, it requires more cognitive effort to evaluate that the social support category does not occur on a given line.

Table 2: Times of annotations for selected bursts with 64,452 utterances.

Annotator’s id	Tagset	Total time (h:mm:ss)	Per utterance
1	SocS	68:47:33	3.84 sec.
2	SocS	77:13:49	4.31 sec.
3	ORB	25:51:13	1.44 sec.
4	ORB	22:27:41	1.25 sec.

3 Limitations and Future Work

IRTIS AT comes with its limitations. First, similarly to other tools, we developed IRTIS AT for a specific task that has arisen. While our tool can be adapted to other tasks that include instant messaging or conversational data, our solution may be unsuitable for other tasks, such as annotating medical records.

Second, in the current version, some functionality has to be solved manually or outside the tool (uploading and anonymizing data, generating bursts for annotation, specifying tags and tagset, detecting disagreements between annotators). In addition, the application lacks more sophisticated authentication. Therefore, future versions should include user accounting with roles (annotator, supervisor, and data manager), store data directly in the database instead of .json files, and allow users to automatically divide and anonymize uploaded data.

Additional functionality that has to be included is automatic IAA calculation and creating bursts of conversations for annotation based on specified rules (e.g., according to the number of lines or conversations) using GUI. Also, the specific burst for supervisors could be available directly in the tool (e.g., all conversations where the concrete annotator puts a question mark as an additional tag).

4 Conclusions

In this paper, we presented a user-friendly annotation tool that allows users to annotate texts from online communicators like Messenger or WhatsApp efficiently. The effectivity is ensured, e.g., due to a well-arranged user interface or involving key pressing to fast annotation of utterances. It also gives annotators and supervisors valuable feedback about how many annotations were done and left, as well as the time spent annotating given bursts of data. The tool has been used in practice for annotating social support and risky behavior in anonymized data of adolescents with sufficient results [14]. Such data can be practically usable in many applications like chat-bots or parental control applications provided by social networking sites.

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

References

1. Cejuela, J.M., McQuilton, P., Ponting, L., Marygold, S., Stefancsik, R., Millburn, G., Rost, B.: tagtog: interactive and text-mining-assisted annotation of gene mentions in plos full-text articles. *Database : the journal of biological databases and curation* **2014**, bau033 (01 2014). <https://doi.org/10.1093/database/bau033>
2. Elavsky, S., Blahošová, J., Lebedíková, M., Tkaczyk, M., Tancos, M., Plhák, J., Sotolář, O., Šmahel, D., et al.: Researching the links between smartphone behavior and adolescent well-being with the future-wp4 (modeling the future: understanding the impact of technology on adolescent's well-being work package 4) project: protocol for an ecological momentary assessment study. *JMIR Research Protocols* **11**(3), e35984 (2022)
3. HumanSignal, Inc.: Label and annotate data. <https://labelstud.io/> (2023), [Online; accessed 25-October-2023]
4. John Snow Labs Inc.: NLP Annotation Lab: Free No Code AI Platform. <https://nlp.johnsnowlabs.com/> (2023), [Online; accessed 25-October-2023]
5. Labelbox, Inc.: Labelbox | Data-centric AI Platform for Building & Using AI. <https://labelbox.com/> (2023), [Online; accessed 25-October-2023]
6. Meta Platforms, Inc.: Facebook: Download a copy of your information on Facebook. <https://www.facebook.com/help/212802592074644> (2023), [Online; accessed 25-October-2023]
7. Meta Platforms, Inc.: WhatsApp: How to save your chat history - Help center. <https://faq.whatsapp.com/1180414079177245> (2023), [Online; accessed 25-October-2023]

8. Montani, I., Honnibal, M.: Prodigy: A modern and scriptable annotation tool for creating training data for machine learning models, <https://prodi.gy/>
9. Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., Liang, X.: doccano: Text annotation tool for human (2018), <https://github.com/doccano/doccano>, software available from <https://github.com/doccano/doccano>
10. Neves, M., Seva, J.: Annotationsaurus: a searchable directory of annotation tools. arXiv preprint arXiv:2010.06251 (2020)
11. Perry, T.: Lighttag: Text annotation platform (2021)
12. Plhák, J., Tkaczyk, M., Lebedíková, M.: Detecting online risks and supportive interaction in instant messenger conversations using czech transformers. (2021)
13. Plhák, J.: Irtis annotation tool. <https://gitlab.fi.muni.cz/irtis/irtis-annotator> (2023)
14. Plhák, J., Sotolář, O., Lebedíková, M., Šmahel, D.: Classification of adolescents' risky behavior in instant messaging conversations. In: International Conference on Artificial Intelligence and Statistics. pp. 2390–2404. PMLR (2023)
15. Sotolář, O., Plhák, J., Lebedíková, M., Tkaczyk, M., Šmahel, D.: Constructing datasets from dialogue data. RASLAN 2022 Recent Advances in Slavonic Natural Language Processing p. 131 (2022)
16. Sotolář, O., Plhák, J., Šmahel, D.: Towards personal data anonymization for social messaging. In: International Conference on Text, Speech, and Dialogue. pp. 281–292. Springer (2021)
17. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 102–107. Association for Computational Linguistics, Avignon, France (Apr 2012), <https://aclanthology.org/E12-2021>

Five Years of Language Services

Zuzana Nevěřilová

Natural Language Processing Centre
Faculty of Informatics
Botanická 68a, Brno, Czech Republic

Abstract. The paper analyzes the usage and patterns observed within Language Services, an aggregation website offering various APIs for natural language processing tasks. Over five years, logs for eight services were collected, allowing for a detailed investigation into the utilization of individual services. Overall, the APIs were used nearly 80 thousand times. The paper focuses on tracking service-specific usage, identifying common trends, detecting potential misuse, and examining error occurrences. The findings provide insights for possible service improvements and future enhancements.

Keywords: declension, tagging, topics, API, log analysis

1 Introduction

Language Services is an aggregation website that provides APIs¹ for various NLP tasks. It was launched in April 2018 without any wide publicity. This paper presents observations from the API logs to see how the services were used. We focused on the number of requests, the number of requests per IP address, and how appropriate the use was. We also discovered that some services did not work for particular inputs or periods.

Section 2 provides an overview of the website usage, and Section 3 describes individual services and observations about their use. Section 4 shows what IP addresses called the services. It can be seen that many users used only one or two services, while others tried all services with a small number of requests. Section 5 summarizes observations of individual services and proposes further improvements.

2 Language Services

The Language Services² provides 13 APIs for Czech and 2 APIs for English. These are:

1. majka – morphological analysis,
2. logic – intensional logical analysis,
3. diacritics – diacritic restoration,
4. inflection – inflection of words,

¹ Application Programming Interface

² <https://nlp.fi.muni.cz/languageservices/>

5. topics – find topics in text, 6. phrases – extraction of (sub)phrases, 7. polite – detection of rude words, 8. vocative – generator of vocative forms, 9. sholva – shallow ontology for Czech words, 10. gen – word forms generator, 11. get location – find location names in text, 12. declension – declension of noun phrases, 13. tagger – tagging of Czech and English, 14. hello – Example service

Unfortunately, we do not have log files for all services; however, for the majority, we do. We investigate eight logs of “real” services (we omit the Hello service); the tagger service is investigated in 3.4 for both languages.

3 Usage Statistics by Service

The Language Services were launched in March 2018 with declension, tagger, polite, diacritics, vocative, get location, topics. The majka service was added in June 2018, gen, logic, and phrases services were added in Fall 2019. We collected usage statistics for the eight services with logs as shown in Table 1. The unknown service means users requested a non-existing service.

Table 1: Language Services usage statistics

service name	requests
declension	10,676
diacritics	12,312
get location	1,034
hello	422
majka	5,673
polite	1,649
tagger	38,759
topics	7,607
unknown	25
vocative	1,071
total	79,228

There were periods when the services were not fully functioning. In Table 2, we provide an overview of error types and the error’s last occurrence. It seems that some errors were fixed meanwhile. Since we log the inputs, the error logs will serve for debugging the services.

3.1 The diacritics service

The service restores diacritics in Czech texts. Since there is massive ambiguity in words without diacritics (e.g., “muzu” might be a form of “můžu” (*I can*), “múzu” (*source of inspiration*), or “mužů” (*man*)), the method is based on n-gram frequencies in the corpus [2]. The diacritic restoration is much more accurate within a context that can reduce the ambiguity.

Table 2: Number of type of errors in the requests

service name	# of errors	error types	last occurrence
declension	3	UnicodeEncodeError UnicodeDecodeError	2022-05-08
diacritics	10	UnicodeEncodeError AttributeError	2023-05-24
polite	11	sre constants (regex error)	2020-03-22
tagger	8	NameError	2023-05-25
topics	681	NameError, TypeError	2021-03-06
vocative	2	IndexError	2023-09-22

The service was used 12,312 times; after filtering out the example requests, the number dropped to 11,869. Next, we filtered out 16 requests with invalid input (missing the text parameter). The number of unique requests was 10,292. The service usage is shown in Figure 1

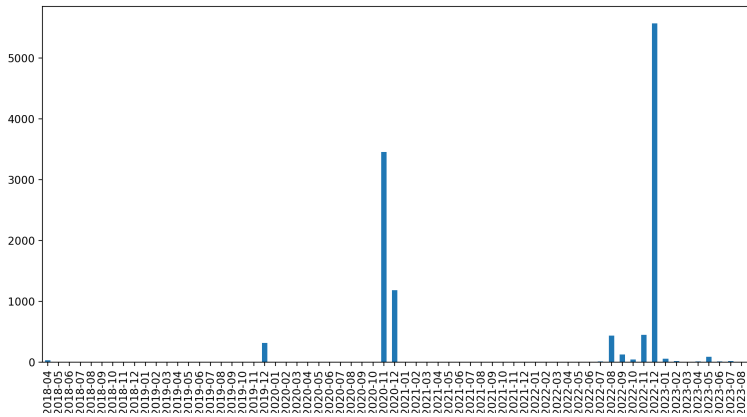


Fig. 1: Usage of the diacritics service (requests without example requests)

In November and December 2020, the service was called 3,236 times by Slovak IP addresses (*orange.sk* and *soitron.sk*). The service was called with single Czech words (e.g., “*predlozim*” (*I will present*), “*jeste*” (*still*), “*strelec*” (*sniper*)).

The highest peak occurred in December 2022, when various IP addresses at *amazonaws.com* used the service more than 5,000 times. It is unclear why the service was used since the inputs contained single Czech words with (!) diacritics.

Overall, the service was mainly misused. Using the service for single words leads to less accurate results, so we should consider providing a disclaimer.

3.2 The declension service

The service for declension provides word form for single nouns or noun phrases. The input is the text and its input case, desired output case, and output number (the same if not provided). The function of the method is described in detail in [1].

The declension service was called 10,676 times; however, 4,947 requests exceeded the daily limit. After filtering out the example requests, there were 6,142 requests. We also filtered out 99 invalid requests, such as missing parameters. The service logs contained 2,655 unique requests.

In 2018-04, the service was tested (480 requests) and used by an IP address at `ncr.com` (275 times), apparently for testing purposes. The service usage is in Figure 2.

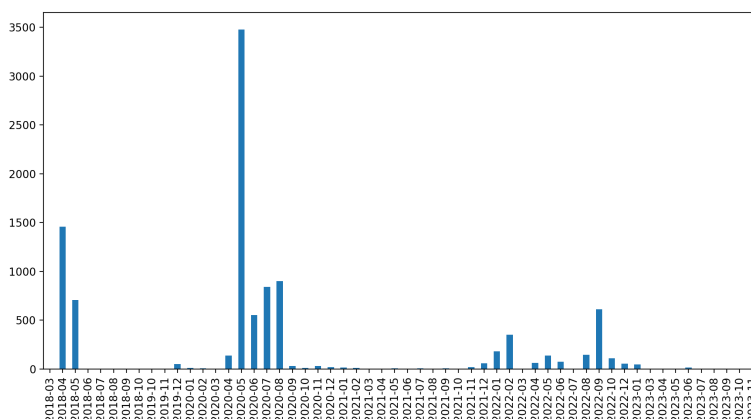


Fig. 2: Usage of the declension service

In May and June 2020, the service was called by IP addresses from Slovakia `slavconet.sk` and `t-com.sk` (666 times). In July and August 2020, the service was used by IP addresses from `shawcable.net` (448 requests), `rogers.com` (557 requests), and `tmcz.cz` (333 requests). The requests from `tmcz.cz` contained noun phrases apparently from newspapers converted to locative case (e.g., “mezinárodní filmový festival” (*International Film Festival*), “britská královská rodina” (*British royal family*), “ministr zahraničí Tomáš Petříček” (*Tomáš Petříček, Minister of Foreign Affairs*), “zmizelý Kim Čong-un” (*the missing Kim Jong Un*)). The requests from `rogers.com` contained complete declension of single words, primarily numerals (e.g., all forms of the word “patnáctý” (*fifteenth*)). The requests from the `slavconet.sk` IP addresses were about the locative case of Czech town and village names (sorted alphabetically). The requests from

t-com.sk aimed to obtain genitive cases of job positions (also sorted alphabetically). The inputs were single words such as “učitel” (*teacher*), “badatelka” (*researcher*), “basista” (*bass player*), “dřevostavatelka” (*woodworker*), “džihadista” (*jihadist*), many of them in their male and female forms (similar to English word pair actor/actress).

A smaller peak appeared in February 2022, when the service was used 212 times by a ssakhk.cz IP address to obtain genitive cases of month names (such as “cerven” – *June* without diacritics). The same IP address used the service 515 times between November 2021 and June 2022.

Another peak in the graph was in 2022-09 when the service was used 503 times by a Vodafone.cz IP address. The service seems to be used for obtaining the base form (nominative) of various phrases from an encyclopedia.

It seems that the service is well understood and not misused. However, for some inputs, the outputs were not correct. These inputs will serve for further improvements of the service.

3.3 The majka service

The Czech morphological analyzer majka [6] is widely used, including its Python binding³. The analyzer provides several modes of operation; the default option is to return tags and lemmata for given input words. Only this option is provided via the Language Services. Also, the original analyzer has several morphological dictionaries⁴, but only the Czech dictionary is provided via the API.

The service was used 5,673 times; after filtering out the example calls, there were 4,893 requests, 3,918 being unique (10 invalid inputs were filtered out). The service usage is in Figure 3.

We investigated the use of the majka service and realized the API is used in a Czechitas Digital Academy project⁵. The project launch and testing are reflected in 2019-12 peak when the service was requested 3,047 times from IP addresses from amazonaws.com, meaning the project was implemented in the AWS Cloud. We did not further investigate the project code. However, we noticed the service was used for individual words in short periods (seconds), even though it can be applied to a list of words.

In April 2020, the service was called 388 times from the IP address ujezd.net. Surprisingly, the service was used to obtain grammar tags for words from the political agenda of the ANO political party.

The highest peak was in 2022-12 when various IP addresses called the service more than 3,000 times at amazonaws.com. The inputs were diverse single Czech words, some being standard (e.g., “diamanty” (*diamonds*), “krabici”

³ <http://pypi.org/project/majka/>

⁴ See <https://nlp.fi.muni.cz/ma/>

⁵ The project *Kategorizace firem podle klíčových slov* from Fall 2019 available at <https://rukkait.blogspot.com/2019/12/v-behaviorurldefaultvml.html>.

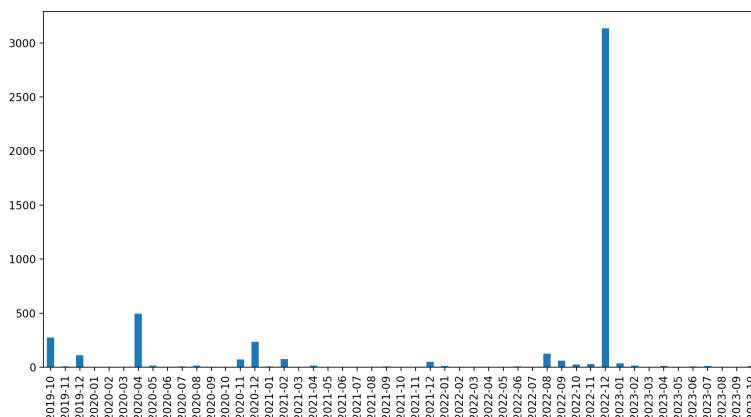


Fig. 3: Usage of the majka service (requests without example requests)

(*box*)), others being non-standard (“sneholak”, probably *snowman*, “kávama”, probably *coffees*).

We checked that most of the time, the service was used for single words (4,336 requests), 145 times, it was used for 2–9 words, 288 times, it was used for 10–100 words, 114 requests containing more than 100 words (note the input limit is 1000 characters). For future improvements, we will inform the users of the possibility of simultaneously processing more than one word.

3.4 The tagger Service

The tagger for Czech is based on the *desamb* tagger [5]. For English, the service uses the TreeTagger implementation [3]. Although TreeTagger supports English, French, German, and Italian, only the English version is implemented in the API service.

The tagger was used 38,759 times; after filtering out example requests, there were 38,083 requests. We removed 472 errors (input errors such as missing parameters and output errors such as no vertical was output by the tagger). Finally, from the 37,611 requests, there were 27,423 unique requests. The tagger was used 153 times for English (with the `lang=en` parameter), and the rest was for Czech. Apart from small samples of English texts (sentences such as “How are you?”), there was one request on January 9–10, 2020, that in 29 requests sent twice the text of Goosey Gazette⁶. The service usage is in Figure 4.

The tagger was used 3,125 times in May 2020 by a Slovak IP address `t-com.sk` for recognizing grammar tags in Czech job position names. The job was run in parallel with the exact requests as the declension service from May 13–16, 2020.

⁶ <https://community.failbettergames.com/t/the-goosey-gazette/18986>

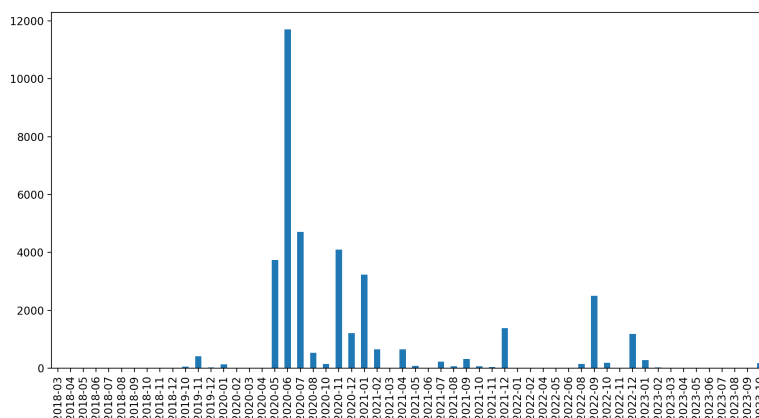


Fig. 4: Usage of the tagger service (requests without example requests)

From May to July 2020, various IP addresses called the service more than 16,200 times from `o2.cz`. The input texts were of a reasonable length (from single sentences to paragraphs), with topics from newspapers, the Bible, and probably fiction.

The 2020-11 peak is caused by 3,247 requests by an `orange.sk` IP address, calling only one-word requests. A similar request was made in December 2020 by another Slovak IP address from `soitron.sk`, sending single-word requests with Czech words sorted alphabetically. The author of this paper uses the tagger service in her teaching, so we are aware of the 556 requests containing parts of the Czech poem *Máj* used in the teaching. The January 2021, December 2021, and December 2022 contain mostly tagging of the poem *Máj*.

The 2022-09 peak is caused by more than 1,200 requests by IP addresses from `o2.cz` that sent texts about literature, history, and politics.

3.5 The topics Service

The service performs a partial syntactic analysis to discover noun phrases. Next, it converts the noun phrases to nominative case (using the underlying application of the declension service). Finally, it scores the noun phrases (by frequency and occurrence of proper nouns).

The service was used 7,607 times. After filtering out the example requests, there were 7,104 requests; after removing 19 invalid requests (missing text parameter) and duplicates, there were 4,981 unique requests. The service usage is shown in Figure 5.

In January and February 2020, the service was used by IP addresses at `tmcz.cz`. The input texts were often too short to output at least one topic. The

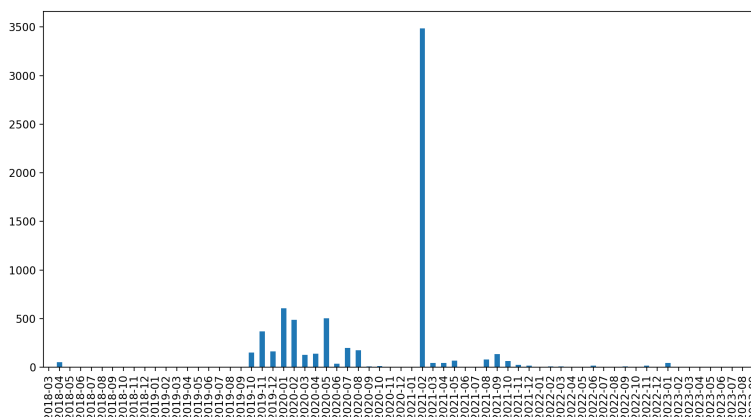


Fig. 5: Usage of the topics service (requests without example requests)

most common topics were “centrum” – *center*, “restaurace” – *restaurant*, “život” – *life*, “kvíz” – *quiz*, “pěkná výstava” – *nice exhibition*.

In 2021-02, the service was requested 3,190 times by a *vodafone.cz* IP address. Almost all the requests contained short texts concerning the “political restart for Czechia” – a concept by Mikuláš Minář (mentioned in the texts) and his political movement “Milion chviliek pro demokracii” (*Million moments for democracy*). The most frequent topics in this set of requests were “politika” (*politics*), “změna” (*change*), “lidé” (*the people*), “svoboda” (*liberty*), “pravda” (*truth*), “demokracie” (*democracy*), most frequent multi-word expressions were “naše země” (*our country*), “noví lidé” (*new people*), “slušní lidé” (*decent people*), “Česká republika” (*the Czech Republic*), “slušná politika” (*decent politics*), “naše děti” (*our children*), “změna politiky” (*change of the politics*), “lepší budoucnost” (*a better future*).

3.6 The `get location` Service

To discover the location mentioned in the input text, the service uses a named entity recognition (NER) implementation for Czech [4].

The service was used 1,034 times; 569 requests differed from the example request. After filtering out one invalid request (without the `text` parameter), there were 352 unique calls. Most of the calls from 2018-04 were for testing purposes.

Apart from the initial testing, the service was not used, so we do not provide a figure. After 2020, the service was used only 38 times. Some of the input texts were quite long (e.g., “Ahoj kde mohu zaparkovat právě stojí na Čápkova 43” – *Hello, where can I park a car near the Čápkova 43*).

The low usage of the service suggests stopping offering it or generalizing it by publishing a state-of-the-art NER service.

3.7 The polite Service

This service is based on a simple list of regular expressions describing rude Czech words.

The service was used 1,649 times. When we excluded the example use, the service was used 1,220 times, from which 613 requests were unique. The service usage can be seen in Figure 6.

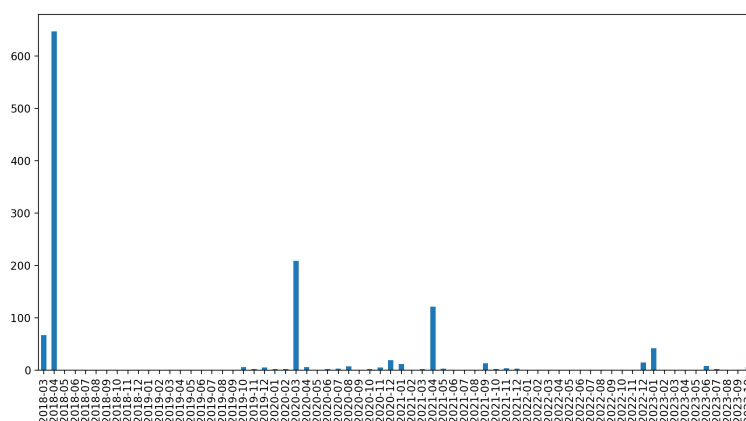


Fig. 6: Usage of the polite service (requests without example requests)

Apart from big testing at the service’s launch, it was used 206 times in 2020-03 by an address at `vodafone.cz`. another peak is in 2021-04, where the service was used 121 times by someone at `amazonaws.com`. Both peak usages were apparently a filter, where, in fact, very few rude words appeared (only “blbec” – *dumb* and its derivatives).

3.8 The vocative Service

The vocative service generates vocative forms for Czech person names. In fact, it is a subset of the declension service. In contrast to the rest of the declension procedures that are based on the `majka` morphological analyzer, the declension of proper nouns is based on separate dictionaries. A similar service exists⁷. The service usage is in Figure 7.

⁷ <https://sklonuj.cz/generator-osloveni/>

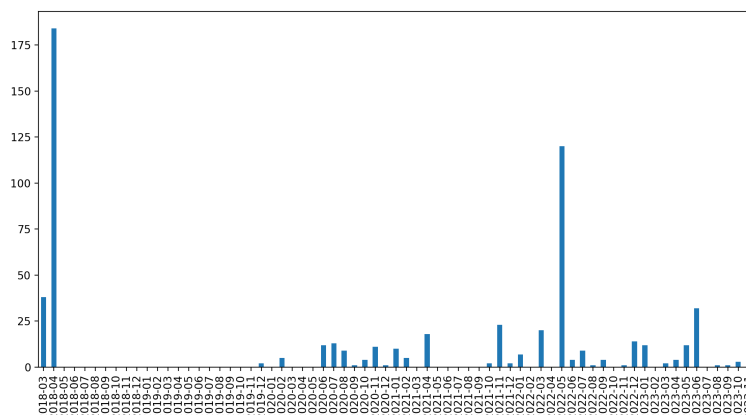


Fig. 7: Usage of the vocative service (requests without example requests)

The service was requested 1,071 times, 589 times with other than example input. There were 2 errors and 187 unique requests. Most of the usage was real person names converted correctly into vocative. Example requests are:

- Dagmar – incorrect vocative Dagmare
- Ivo Václav Hawiger – incorrect vocative Ive Václave Hawigere
- Klaus Mueller – incorrect vocative Klac Muellere
- Pepek Námořník – correct vocative Pepku Námořníku
- Jindřich Mořeplavec – incorrect vocative Jindřichu Mořeplavec
- Jana Malá – correct vocative Jano Malá
- Martínek – correct vocative Martínku
- milada horáková – correct vocative milado horáková
- Admin – correct vocative Admin
- Aleksandra Pavlovna Vysockaja – correct vocative Aleksandro Pavlovno Vysockaja

The usage peaks were at the beginning when the service was launched and tested by 6 different IP addresses, then in 2022-05 when the service was used 117 times by an IP address at `selfnet.cz`.

4 Statistics of IP Addresses

During the observed period, 2,164 different IP addresses used the Language Services. In this Section, we investigated the origins of the requests. Although it is not possible to find real people or organizations from most of the IP addresses, we grouped the IP addresses based on our knowledge. The `amazonaws.com` requests are requests from programs stored in the Amazon AWS Cloud, meaning

someone is using the APIs inside their own programs. A similar situation is with the `googleusercontent.com` IP addresses; these are API calls from Google Colabs used in teaching.

Table 3: Number of requests per IP address

IP address	declension	topics	tagger	majka	get	location	polite	diacritics	vocative
amazonaws.com				634					1016
amazonaws.com		1312					242		
amazonaws.com				642					1002
amazonaws.com									1438
amazonaws.com				1084					892
amazonaws.com				648					994
amazonaws.com				1296					2024
amazonaws.com				602					978
cvut.cz		1502	2						
ssakhk.cz		1030							
soitron.sk				2136	22				1582
orange.sk				6496					6474
t-com.sk		6896		6250					
o2.cz				2446					
o2.cz				1394					
o2.cz				15648					
o2.cz				6692					
o2.cz				1352					
o2.cz				1354					
tmcz.cz	2	1004					6		
tmcz.cz	970	712	252	18		6	6	4	4
vodafone.cz	12	82				860	1178	2	326
vodafone.cz	2	6506	8	2					
vodafone.cz	1228		1346						
vodafone.cz	1384							2	
ncr.com	1018								60
shawcable.net	1260								
rogers.com	1544			2					

To our knowledge, Language Services are used in student projects at Masaryk University. Moreover, the `majka` service is used in the Czechitas Digital Academy project (see Section 3.3). Other schools used the Language Services as well (`cvut.cz` and `ssakhk.cz` is a university and a high school, respectively).

Other IP addresses are owned by Internet providers in Czechia, Slovakia, and worldwide. We cannot conclude anything. Table 3 shows domains of IP addresses that requested Language Services more than 1000 times. It can be clearly seen that `tagger` is the most popular service for Slovak IP addresses, `majka` and `diacritics` are most widely used in other applications. The IP addresses at `o2`

and vodafone might belong to the same subject as both companies use a dynamic IP address assignment.

On the other hand, among the 135 IP addresses that requested more than six different services, only five sent more than 200 requests in total. This indicates another typical behavior – someone uses the APIs for experiments but not for further benefit.

5 Conclusion and Future Work

After five years of running the Language Services, it can be seen the service is known. Since it started as a toy project, there are no standard functions such as the health check or various HTTP codes for various events (e.g., 429 Too many requests). Instead, the service returns HTTP status 200 (OK) and the message inside the response. We plan to improve the service on this technical level.

For individual services, we collected enough data about how they are used. It seems reasonable to explain further what the service is good for to avoid inefficient use. In the future, we will focus on more comprehensible documentation and improvement of individual services (declension, evocative, tagger). Also, other services should store the log information for future usage analysis.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Nevěřilová, Z.: Declension of Czech Noun Phrases. In: Radimský, J. (ed.) *Actes du 31e Colloque International sur le Lexique et la Grammaire*. pp. 134–138. Université de Bohême du Sud à České Budějovice (République tchèque), České Budějovice (2012)
2. Rychlý, P.: Czaccent - simple tool for restoring accents in Czech texts. In: Aleš Horák, P.R. (ed.) *6th Workshop on Recent Advances in Slavonic Natural Language Processing*. pp. 15–22. Tribun EU, Brno (2012), <https://nlp.fi.muni.cz/raslan/2012/paper14.pdf>
3. Schmid, H.: TreeTagger (2014), <http://hdl.handle.net/11372/LRT-323>, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
4. Sedlář, L.: Hybridní systém pro detekci pojmenovaných entit v českém textu. Diploma thesis, Masaryk University, Faculty of Informatics (2014), <https://is.muni.cz/th/rij93/>
5. Šmerk, P.: Unsupervised learning of rules for morphological disambiguation. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *Text, Speech and Dialogue*. pp. 211–216. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
6. Šmerk, P.: Fast morphological analysis of Czech. In: *Proceedings of the Raslan Workshop 2009*. Masarykova univerzita, Brno (2009), <https://nlp.fi.muni.cz/raslan/2009/papers/13.pdf>

Document Visual Question Answering with CIVQA

Czech Invoice Visual Question Answering Dataset

Šárka Ščavnická , Michal Štefánik , and Petr Sojka 

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
527352@mail.muni.cz

Abstract. Applications of document processing become increasingly popular across multiple industries, resulting in a growing amount of research on the applications of artificial intelligence in document processing (Document AI). This paper focuses on a subtask of Document AI, Document Visual Question Answering (DVQA), recently getting well-deserved attention thanks to its universality. However, the limited availability of data sources for languages outside English restrains the applicability of DVQA in non-English languages.

For this reason, we created the CIVQA (Czech Invoice Visual Question Answering) dataset covering 15 entities of financial documents, consisting of more than 6,000 invoices in the Czech language.

We used the CIVQA dataset to create the first-of-its-kind DVQA models specifically tailored for applications to Czech documents. Striving to create DVQA models able to generalize, we specifically evaluate our models on the entities not covered in the training mix and find that multilingual LayoutLM models are able to respond to questions about previously unseen entities substantially more accurately than other models.

The CIVQA dataset and experiment observations offer new opportunities for Document AI in the Czech Republic, with potential applications in research and commercial fields.

Keywords: Question Answering, Visual Question Answering, Document Visual Question Answering, Czech Invoice Visual Question Answering Dataset

1 Introduction

Document AI is transforming how businesses and organizations process, store, and analyze vast amounts of data. [26] have stated that The Big Four accounting firms (Deloitte, Ernst & Young (EY), PricewaterhouseCoopers (PwC), and Klynveld Peat Marwick Goerdeler (KPMG)) have launched their own Document AI systems. These systems are able to automatically recognize data from Visually Rich Documents, enter invoices into the systems, and generate financial reports.

Document Visual Question Answering (DVQA) is a subgroup of Document AI. The task of question answering is usually combined with the use of the Large

Language Models (LLM). Demand after these systems is most visible in the domain of office documents like invoices [19].

For creating suitable models for DVQA, it is essential to have appropriate learning data [11,21]. While several English datasets exist, no DVQA datasets exist for the Czech language. To address this gap, we have created the first dataset in the Czech language, focused on invoices. We conducted experiments to evaluate the quality and robustness of the CIVQA dataset and DVQA models trained on CIVQA. Our results demonstrate that CIVQA-trained models can to a certain extent generalize to unseen question types, and the robustness of resulting DVQA models can be further supported by using multilingual base models and mixing of CIVQA datasets with existing datasets in English.

2 Background

This section gives a brief theoretical introduction to the Visually Rich Documents and Document Visual Question Answering.

2.1 Visually Rich Documents

Visually Rich Documents (VRD) contain documents whose semantic structure is not determined only by the text but also by the layout and visual elements of the documents. These visual elements are, for example, typesetting formats, tables, and figures. Invoice is an example of VRD; its layout pieces of information are crucial for the overall understanding because they are usually split into several sectors. [9]

2.2 Document Visual Question Answering

Document AI can be divided into four groups: Document Layout Analysis, Visual Information Extraction, Document Visual Question Answering, and Document Image Classification [9]. In this paper, we will be focusing on the third part: Document Visual Question Answering.

The Question Answering (QA) systems are tools for retrieving specific information that some users have requested. One of the most significant features is that the Document Visual Question Answering systems can retrieve these pieces of data from the VRDs. [27] The usual QA systems have two inputs: the first is the question, and the second is a document or collection of multiple documents, where we search for the answer.

The question-answering systems have evolved over time. First, there were purely text-based systems; for example, these systems worked with Wikipedia articles and looked for factual answers. The BERTmodel [2], fine-tuned on the SQuAD dataset [18], is an example of a textual QA model.

Subsequently, Visual question-answering (VQA) models began to emerge. Antol et al. [1] define VQA as an artificial intelligence technology that enables a computer system to answer questions about an image. VQA combines natural

language processing, object recognition, and computer vision to interpret the content of an image and provide answers based on its understanding.

Document Visual Question Answering [10] seeks to obtain knowledge from documents through answering questions. The asked questions may relate to different parts of the examined document, not only the text part; for example, they may refer to inserted images, tables, and forms, but they may also refer to the overall arrangement of the text. Therefore, for Document VQA, we need to incorporate the detection of scene objects and an understanding of the document's layout and the relations between different parts of the layout. Due to their ability to work with VRDs, Document VQA popularity constantly increases across different fields. For example, they can help process invoices and other documents in the financial sector.

Only a small amount of Document VQA datasets have been created recently, primarily in English. These datasets typically feature web pages, scanned documents, born-digital documents, as well as pages sourced from textbooks or posters.

Nowadays, the most popular dataset for Document VQA is DocVQA [17]. This dataset comprises several documents from the UCSF Industry Documents Library [23], which also includes invoices. The documents are either born-digital, scanned, handwritten, or typewritten from 1960 to 2000.

2.3 Models

The models from the LayoutLM family play an essential role in Document AI, mainly because during pre-training, they combine both the visual part of the document and its textual part. [9] Hence, they are improving the performance of Document AI models.

In this paper, we have focused on five different versions of the LayoutLM model family: LayoutXLM (Layout Cross-Lingual Language Model) [25], LayoutLMv2 [24], LayoutLMv3 [14], Impira QA (Impira model for Visual Question Answering) [16], and Impira Invoice (Impira for Invoices) [15]. Impira models are the finetuned versions of the LayoutLM models.

3 CIVQA Dataset

Presently, coverage of non-English models for Document Visual Question Answering is lacking. For this reason, we have created the first Czech dataset for document question-answering, called the *CIVQA dataset*.

The CIVQA dataset consists of 6,849 invoices, which were obtained from public sources. Over these invoices, we focused on 15 different entities, which are crucial for processing the invoices. We included each entity in at least one of these four groups: numeric, textual, pattern, and shape. The difference between the pattern and shape is that patterns are entities like QR codes, which do not contain words or numbers but have some *visual* pattern. The shape group is for

Table 1: Entities' categories

Entity	Numeric	Textual	Pattern	Shape
Invoice number	X			
Variable symbol	X			
Specific symbol	X			
Constant symbol	X			
Bank code	X			X
Account number	X			X
ICO	X			X
Total amount	X			
Invoice date	X			X
Due date	X			X
Name of supplier		X		
IBAN	X	X		X
DIC	X	X		X
QR code			X	
Supplier's address		X		

id string · lengths	words sequence	answers string · lengths	bboxes sequence	answers_bboxes sequence	questions string · lengths	image string · lengths
9	9	1 72			4 35	36 36
420000000	["12008626", "FAKTÚRA", "(FV)", "Strana", "1",...]	Rosinská cesta 13 010 08 Žilina	[[78.69774919614147,...]	[[15.434083601286174...	Jaká je adresa dodavatele?	f8f55985f6a82596baa43a72543fa4e4
420000001	["12008626", "FAKTÚRA", "(FV)", "Strana", "1",...]	Rosinská cesta 13 010 08 Žilina	[[78.69774919614147,...]	[[15.434083601286174...	Kde sídlí dodavatel	f8f55985f6a82596baa43a72543fa4e4

Fig. 1: CIVQA pre-encoded dataset

input_ids sequence	bbox array 2D	attention_mask sequence	image array 3D	start_positions int64	end_positions int64	questions string · lengths	answers string · lengths
[0, 4422,...]	[[0, 0, 0, ...]	[1, 1, 1, 1, 1, 1, 1, 1, ...]	[[[140, 145, 147, 149, 151, 153, 156, 156, 157, 159, 160, 163, 163, 161, 166, 169, 171, 170, 171,...]	5 510	7 510	4 35	1 72
[0, 4422,...]	[[0, 0, 0, ...]	[1, 1, 1, 1, 1, 1, 1, 1, ...]	[[[140, 145, 147, 149, 151, 153, 156, 156, 157, 159, 160, 163, 163, 161, 166, 169, 171, 170, 171,...]	91	100	Jaká je adresa dodavatele?	Rosinská cesta 13 0: Žilina
[0, 119950,...]	[[0, 0, 0, ...]	[1, 1, 1, 1, 1, 1, 1, 1, ...]	[[[140, 145, 147, 149, 151, 153, 156, 156, 157, 159, 160, 163, 163, 161, 166, 169, 171, 170, 171,...]	88	97	Kde sídlí dodavatel	Rosinská cesta 13 0: Žilina

Fig. 2: CIVQA encoded dataset

entities that have some given rules and constraints like IBAN or DIC. All entities and their groups can be seen in Table 1.

The first step needed for creating the datasets was obtaining annotated invoices. The annotation was provided by a third party in the Intelligent Backoffice (IBO) project [20,13]. The annotations and OCR results were received in a JSON file, which includes several items for each invoice. These items consist of the unique identifier of the image, image path, bounding boxes of all words on the invoice obtained by OCR from the invoice, a list of words corresponding to the respective bounding boxes, and a list of labels corresponding to the fifteen entities, as shown in Table 1. We have also added a label marked "O" to the list

of labels. This label belongs to those words that, during annotation, were not assigned any entity.

This JSON is further processed to add questions and gather answers. We have created at least three questions per entity. Each entity corresponds to an answer that we are looking for. With the help of these questions, we covered an extensive range of possibilities a user can ask about an entity. Examples of questions created for the invoice number entity are: Jaké je číslo faktury? (What is the invoice number?), Pod jakým číslem je vedena faktura? (Under which number is the invoice kept?), Číslo faktury? (Invoice number?), Jaké je označení faktury? (What is the label on the invoice?).

There are two types of CIVQA datasets. The first was created using Tesseract OCR [22], and the second was created with EasyOCR [12]. At the same time, both datasets have two versions. The first one is suitable for further use and subsequent adaptation of this dataset for other models. The part of the dataset can be seen in Figure 2 on the facing page. It contains words, respective bounding boxes, image names, questions, and answers. There are no labels, as these labels may vary for different models for document visual question answering, though this dataset is ready to be encoded for future models.

Figure 2 displays the second dataset, which is ready for training on selected models. The images in this dataset are resized to 224×224 , and it is established that they have the correct order of color channels. Due to the resizing of the images, it was also necessary to recalculate all the bounding boxes. The words and bounding boxes are transformed into token-level parameters like `input_ids`, `attention_mask`, `token_type_ids`, and `bbox`. The processor adds special tokens ([CLS] and [SEP]) to separate questions from word tokens. For the chosen LayoutLM model, we also need labels; for these models, the label consists of starts and end positions, indicating which token is at the start and which token is at the end of the answer. Based on this, we can find the correct answer to our questions. The dataset also contains the textual answer, which can be used for verification if the model is predicting the correct answer. It is important to note that these datasets contain errors created by OCR retrieval of incorrect letters and mistakes made by annotators.

CIVQA datasets are available as CIVQA TesseractOCR Dataset [7], CIVQA TesseractOCR LayoutLM Dataset [8], CIVQA EasyOCR Train Dataset [5], CIVQA EasyOCR Validation Dataset [6], CIVQA EasyOCR LayoutLM Validation Dataset [4], CIVQA EasyOCR LayoutLM Train Dataset [3].

4 Experiments

In this section, we introduce experiments done with CIVQA datasets. We fine-tuned the models from Section 2.3 on CIVQA datasets in order to find out which OCR method is the best for Czech Document VQA. In the second part of the experiments, we evaluated the robustness of the resulting models on unseen types of questions.

4.1 The OCRs and the CIVQA Dataset

The objective of this experiment was to determine the impact of using different OCRs on the quality of document visual question-answering systems and find the best OCR for future experiments. The model’s quality is determined by its ability to return a prompt identical to the answer from the dataset. The better the model, the better the precision it achieves. For this experiment, we focused on all 15 entities throughout all invoices. Table 2 shows the measured results for both OCR frameworks. We see that the best results were obtained by fine-tuned models on the dataset used by Tesseract OCR, which can recognize more languages than EasyOCR. Furthermore, LayoutXLM achieves the best results for both types of datasets.

Table 2: CIVQA results: comparison of Tesseract and EasyOCR frameworks by Precision, Recall, and F1 score.

Model	Tesseract			EasyOCR		
	Prec	Recall	F1	Prec	Recall	F1
LayoutXLM	0.7422	0.7117	0.7079	0.6636	0.6633	0.6455
LayoutLMv2	0.6917	0.6750	0.6634	0.6323	0.6129	0.6011
LayoutLMv3	0.6989	0.6382	0.6410	0.6370	0.6164	0.6065
Impira QA	0.6773	0.6291	0.6313	0.6373	0.6015	0.5984
Impira Invoice	0.6948	0.6440	0.6434	0.6345	0.6019	0.5962

As stated, the LayoutXLM is the overall best model for the Czech document visual question-answering task. This model is unique because it was trained using a multilingual dataset consisting of these languages: Chinese, Japanese, Spanish, French, Italian, German, and Portuguese [25]. Even though it was not trained on the Czech dataset, the languages with diacritic (Spanish, French, German, Portuguese, Italian) may have helped the LayoutXLM to work better than other models (trained only on English data).

We also delved into exploring the performance of individual questions. In Figure 3 on the next page, it can be seen how many percentages of questions were answered correctly for the best model from Table 2. Individual questions are separated by color and assigned to individual entities.

The success of individual entities has been observed to be influenced by the entity groups that were defined in Table 1, as we can see in the percentage results for individual questions in Figure 3 on the next page. Entities with a specific or numerical shape were more successful than purely textual entities without a specific shape. The supplier’s name and the supplier’s address are the entities with one of the lowest success rates, and these are also the only purely textual entities. Below them is the QR code entity, which is neither textual nor number and is spread on multiple lines, which is a problem when predicting starting and ending positions.

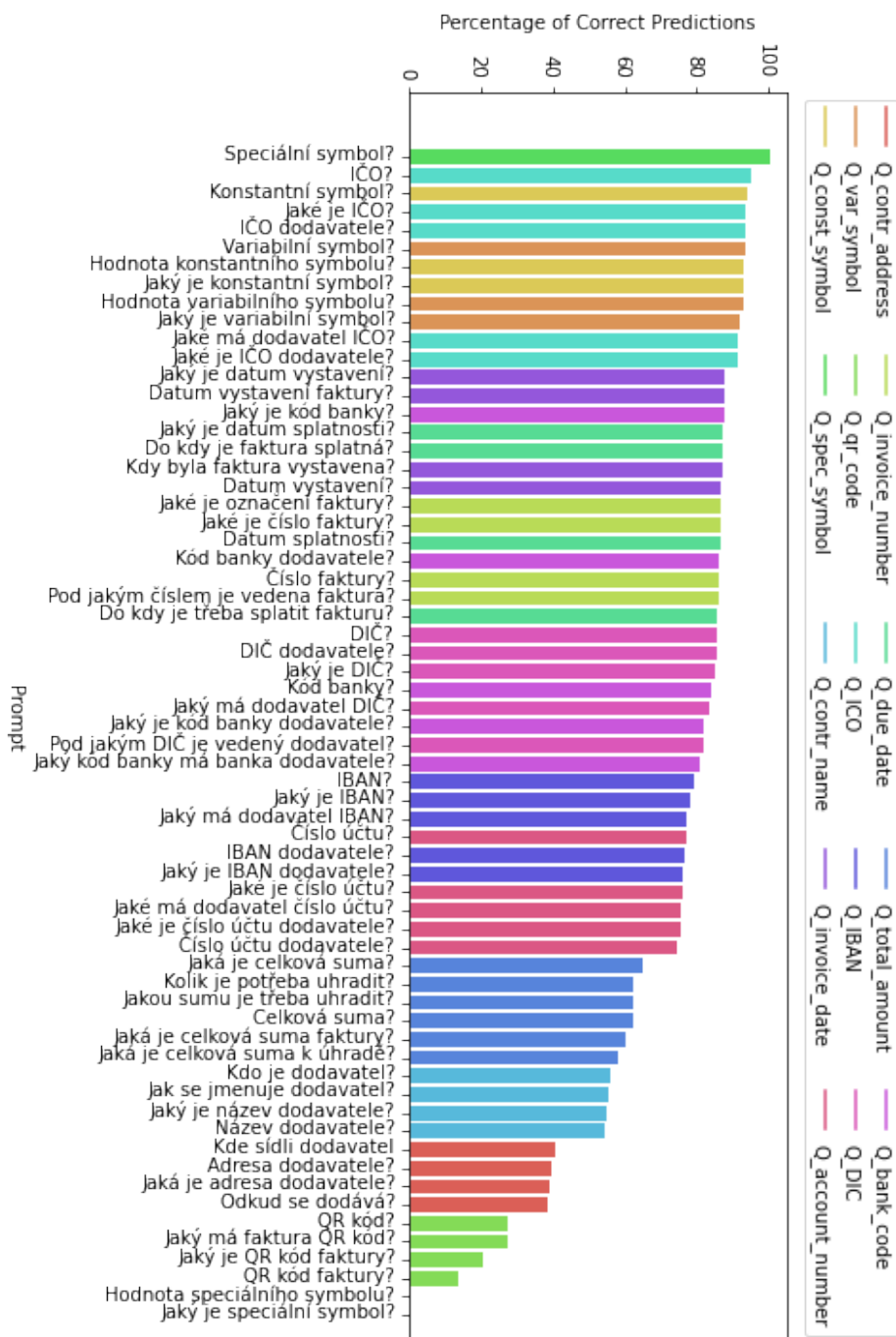


Fig. 3: Validation dataset of CIVQA Tesseract OCR: LayoutXLM model success rate by individual question percentage.

The entity with the identifier *ICO*, which is a numeric entity with a given shape, performed the best in the evaluation. Although in the first place, with 100% success, we have a question focused on the *Specific symbol* a numeric entity. However, this happened for one out of three types of questions for this entity. All other questions related to this entity achieved a 0% success rate. Based on this, we can not say this is the best-performing entity when *ICO* had good results for every question. It should be also noted that the *Specific symbol* entity was presented less than the other entities in the obtained invoices because this is not that much used on the invoices.

4.2 CIVQA and unseen types of questions

In this set of experiments, our focus was on developing a practical and robust solution for unseen entities. We would like to create a model that could be used on new entities, and in that case, it could be more beneficial for users. For this task, we have separated the CIVQA_TesseractOCR dataset into two datasets. One is for unseen entities with five entities, and the other is for known entities. We choose these five entities (invoice number, *ICO*, supplier's address, IBAN, due data), in the way we would cover the most different types of entities, based on the Table 1 on page 26.

In the following subsections, we will present various experiments where we observed how the models behave on unknown entities. Initially, we trained individual models from Section 2.3 on a new dataset of ten known entities and verified their success on unknown entities. Subsequently, we tried to improve their success with various attempts.

Table 3: CIVQA results: comparison of models when handling unknown entities

Model	Baseline			Known data			DocVQA + Known data		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
LayoutXLM	0	0	0	0.1920	0.0413	0.0582	0.3731	0.2163	0.2465
LayoutLMv2	0	0	0	0.0343	0.0270	0.0261	0.0665	0.0334	0.0279
LayoutLMv3	0	0	0	0.1022	0.0341	0.0456	0.1504	0.0455	0.0611
Impira QA	0	0	0	0.1512	0.0455	0.0652	0.2326	0.0895	0.1148
Impira Invoice	0	0	0	0.1360	0.0530	0.0724	0.2226	0.0807	0.1063

In Table 3, we have presented the results of our experiments on unknown entities. The first column (baseline) contains the results for each model without training them on known values. Neither model was successful and failed to correctly predict any entity, resulting in a precision, recall, and F1-score of 0.

The second column (known data) represents the results of the models, which were finetuned with the dataset of ten known entities. This is the first introduction of the Czech language to these models, and we can see that the models were now more successful with predicting some entities.

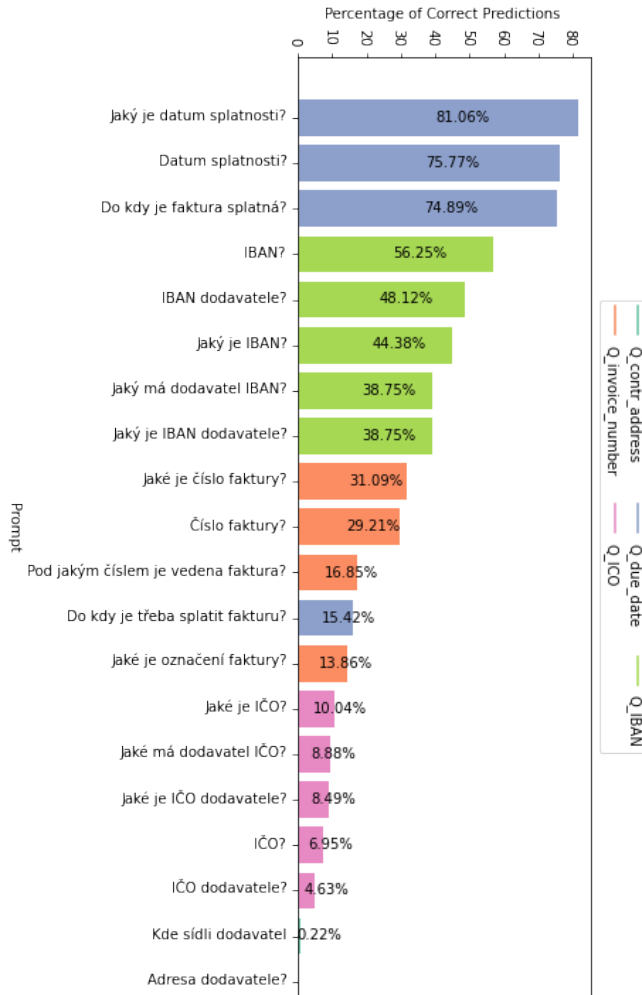


Fig. 4: Validation dataset of CIVQA unknown entities: LayoutXLM model success rate by individual question percentage fine-tuned on DocVQA plus CIVQA known dataset.

The third column (DocVQA + known data) shows the outcomes obtained from fine-tuning models on a dataset of known entities concatenated with the DocVQA dataset, which is currently one of the best datasets for DVQA. We can see that even though the DocVQA only consists of invoices in English, it has improved the score on unknown entities. The LayoutXLM model consistently achieved the best results.

For the best model, we have created a graph of the success of individual questions, where we will take a closer look at the success of each entity. This

graph can be seen in Figure 4. In this case, entities with some clear structure and shape are also more successful. The entity known as ‘due date’ (a numeric entity with a specific format) has achieved first place with an outstanding success rate. This entity is also similar to the invoice date entity, which is in the known entities dataset. Based on this, we can claim that if there is a similar type of entity to the unknown in the dataset of known values, the model will correctly predict this unknown but similar entity.

4.3 Effect of introduction of a small number of unknown entities

In this experiment, we have tried introducing a small amount of unknown data to the trained models. We choose 5% and compare these results with the results obtained without the introduction of unknown data in order to see how it affects the models.

In Table 4, we compare results obtained from various models. The first column results were not exposed to any unknown data, though they were fine-tuned on known data. We then fine-tuned these models with 5% of unknown data and compared the results, which can be seen in the third column. Lastly, we will compare the models from the first column, but it was further fine-tuned on the known dataset concatenated with 5% of unknown data. This experiment aimed to evaluate how different amounts of data, even those already known to the models, could improve the overall results.

According to Table 4, we can see that no other model has not surpassed LayoutXLM. Fine-tuning with a small part of unknown entities showed noticeable improvement in all models. However, after using the concatenated dataset of known entities with 5% of unknown entities, LayoutXLM did not obtain as much of an improvement as other models.

Table 4: CIVQA results: Comparing results on baseline models, then models trained on a 5% subset of unknown entities and then models fine-tuned on the concatenation of known dataset with a subset of 5% unknown entities.

Model	Known data			5% of unknown			Known + 5% unknown		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
LayoutXLM	0.1920	0.0413	0.0582	0.7002	0.6594	0.6617	0.7069	0.6693	0.6700
LayoutLMv2	0.0343	0.0270	0.0261	0.5944	0.5154	0.5192	0.6223	0.5726	0.5755
LayoutLMv3	0.1022	0.0341	0.0456	0.5793	0.5125	0.5254	0.6344	0.5528	0.5631
Impira QA	0.1512	0.0455	0.0652	0.6186	0.5356	0.5466	0.6318	0.5487	0.5670
Impira Invoice	0.1360	0.0530	0.0724	0.5999	0.5255	0.5369	0.6353	0.5577	0.5681

5 Conclusion

This paper introduces the CIVQA datasets, which are opening new doors in the field of Document VQA in the Czech language. We have discovered that numeric answers obtained better results than purely textual ones. Furthermore, we have shown that combining the CIVQA dataset with another DVQA dataset can improve the robustness of DVQA on unseen entities.

Acknowledgements. We acknowledge the support of grant Intelligent Back Office, project number CZ.01.1.02/0.0/0.0/21_374/0026711.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual Question Answering. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2425–2433 (2015). <https://doi.org/10.1109/ICCV.2015.279>
2. Chan, B., Möller, T., Pietsch, M., Soni, T.: Hugging Face (2022), <https://huggingface.co/deepset/roberta-base-squad2>
3. CIVQA EasyOCR LayoutLM Train Dataset, https://huggingface.co/datasets/fimu-docproc-research/CIVQA_EasyOCR_LayoutLM_Train, accessed 2023-11-21
4. CIVQA EasyOCR LayoutLM Validation Dataset, https://huggingface.co/datasets/fimu-docproc-research/CIVQA_EasyOCR_LayoutLM_Validation, accessed 2023-11-21
5. CIVQA EasyOCR Train Dataset, https://huggingface.co/datasets/fimu-docproc-research/CIVQA_EasyOCR_Train, accessed 2023-11-21
6. CIVQA EasyOCR Validation Dataset, https://huggingface.co/datasets/fimu-docproc-research/CIVQA_EasyOCR_Validation, accessed 2023-11-21
7. CIVQA TesseractOCR Dataset, <https://huggingface.co/datasets/fimu-docproc-research/CIVQA-TesseractOCR>, accessed 2023-11-21
8. CIVQA TesseractOCR LayoutLM Dataset, <https://huggingface.co/datasets/fimu-docproc-research/CIVQA-TesseractOCR-LayoutLM>, accessed 2023-11-21
9. Cui, L., Xu, Y., Lv, T., Wei, F.: Document AI: Benchmarks, Models and Applications (2021). <https://doi.org/10.48550/arXiv.2111.08609>
10. Ding, Y., Huang, Z., Wang, R., Zhang, Y., Chen, X., Ma, Y., Chung, H., Han, S.C.: V-Doc: Visual questions answers with Documents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21492–21498. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2022). <https://doi.org/10.1109/CVPR52688.2022.02083>
11. DocILE 2023: ICDAR 2023 Competition and CLEF 2023 Lab on Document Information Localization and Extraction (2023), <https://docile.rossum.ai/>, accessed 2023-10-31
12. EasyOCR, <https://github.com/JaidedAI/EasyOCR>, accessed 2023-08-10
13. Geletka, M., Bankovič, M., Meluš, D., Ščavnická, Š., Štefánik, M., Sojka, P.: Information Extraction from Business Documents: A Case Study. In: Horák, A., Rychlý, P., Rambousek, A. (eds.) Proceedings of the 16th Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2022, Karlova Studánka, Czech Republic, December 9–11, 2022. pp. 35–46. Tribun EU (2022), <https://nlp.fi.muni.cz/raslan/2022/paper18.pdf>

14. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for Document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 4083–4091 (2022)
15. LayoutLM for Invoices, <https://huggingface.co/impira/layoutlm-invoices>, accessed 2023-10-25
16. LayoutLM for Visual Question Answering, <https://huggingface.co/impira/layoutlm-document-qa>, accessed 2023-10-25
17. Mathew, M., Karatzas, D., Jawahar, C.: DocVQA: A Dataset for VQA on Document Images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2200–2209 (Jan 2021)
18. Rajpurkar, P., Jia, R., Liang, P.: Know What You Don't Know: Unanswerable Questions for SQuAD. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the ACL (Volume 2: Short Papers). pp. 784–789. ACL, Melbourne, Australia (Jul 2018). <https://doi.org/10.18653/v1/P18-2124>
19. Rossum raises record \$100 million Series A from General Catalyst to reinvent B2B document communication (2021), <https://rossum.ai/blog/rossum-raises-record-100-million-series-a-from-general-catalyst-to-reinvent-b2b-document-communication/>, accessed 2023-10-31
20. Ščavnická, Š., Štefánik, M., Kadlčík, M., Geletka, M., Sojka, P.: Towards General Document Understanding through Question Answering. In: Horák, A., Rychlý, P., Rambousek, A. (eds.) Proceedings of the 16th Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2022, Karlova Studánka, Czech Republic, December 9-11, 2022. pp. 181–188. Tribun EU (2022), <https://nlp.fi.muni.cz/raslan/2022/paper17.pdf>
21. Šimsa, Š., Šulc, M., Uříčář, M., Patel, Y., Hamdi, A., Kocián, M., Skalický, M., Matas, J., Doucet, A., Coustaty, M., Karatzas, D.: DocILE Benchmark for Document Information Localization and Extraction (2023), <https://arxiv.org/abs/2302.05658>
22. Tesseract Open Source OCR Engine (main repository), <https://github.com/tesseract-ocr/tesseract>, accessed 2023-10-08
23. UCSF Industry Documents Library, <https://www.industrydocuments.ucsf.edu/>
24. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on NLP (Volume 1: Long Papers). pp. 2579–2591. ACL, Online (Aug 2021). <https://doi.org/10.18653/v1/2021.acl-long.201>
25. Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florencio, D., Zhang, C., Wei, F.: LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding (2021). <https://doi.org/10.48550/arXiv.2104.08836>
26. Zhang, Y., Xiong, F., Xie, Y., Fan, X., Gu, H.: The Impact of Artificial Intelligence and Blockchain on the Accounting Profession. IEEE Access 8, 110461–110477 (2020). <https://doi.org/10.1109/ACCESS.2020.3000505>
27. Zitouni, I.: Natural Language Processing of Semitic Languages. Springer (2014). <https://doi.org/10.1007/978-3-642-45358-8>

Can We Detect ChatGPT-generated Texts in Czech and Slovak Languages?

Petr Šigut and Tomáš Foltýnek

Faculty of Informatics, Masaryk University,
Brno, Czech Republic
514530@mail.muni.cz, foltynnek@fi.muni.cz

Abstract. The wide availability of generative AI exacerbates existing threats to society. It would not be easy even for linguists to tell whether the text we are reading was generated by a Large Language Model (LLM) or written by a human.[1]. Researchers have started developing tools that detect AI-generated content [2]. This paper tested how two of these tools, Compilatio [3] and GPT-2 Output Detector [4], performed with Czech, Slovak and English texts. There was only one tool somewhat capable of detecting AI-generated texts: Compilatio. Other tools were designed to work only with English texts. Hence, we also tested whether automatically translating the Czech and Slovak texts to English before uploading them to the detectors would have given any promising results. Ultimately, we showed that the texts generated by ChatGPT4 were less detectable than the texts generated by ChatGPT3.5.

Keywords: ChatGPT, AI-detection, Czech, Slovak

1 Introduction

The launch of ChatGPT in November 2022 impacted many areas of human activity. For example, universities have been concerned with detecting unauthorised content generation [5], researchers are worried about the influx of AI-generated papers and the impact of the use of AI in the medical field [6], others are worried by the rise of AI-generated fake news. These threats create the need for reliable AI detection tools. This is particularly relevant regarding texts generated by AI in languages other than English, as most existing tools are trained to work with English texts primarily.

Several studies compare the performance of AI detection tools [9,8,7]. The study by Chaka [7] performed a test with generated documents from ChatGPT, YouChat and Chatsonic, which were subsequently translated into German, French, Southern Sotho, isiZulu and Spanish. Their test consisted of five AI content detectors: Open AI Text Classifier, Writer, GPTZero, Copyleaks and Giant Language Model Test Room. Still, their findings showcased that only one tool (Copyleaks) could detect some AI-generated documents in German, French and Spanish languages. Overall, the paper concluded that none of the tools are fully ready to detect AI-generated texts in different languages accurately.

Considering the above-mentioned results, this paper seeks to examine whether the state of AI detection has changed, particularly when dealing with languages other than English. It builds upon previous research conducted by Weber-Wulff et al.[9], who tested the performance of 14 tools for detecting AI-generated text in English. This paper examines how accurately the detectors can recognise AI-generated text in Czech and Slovak languages compared to English. It tests two of the best currently publicly available tools: Compilatio and GPT-2 Output detector. The paper focuses on both the texts in their original languages (Czech and Slovak) and investigates whether their subsequent translation to English might deliver different results. Lastly, the paper also examines the extent to which ChatGPT4 and ChatGPT3.5 versions differ in the detectability of the content they produce.

2 Methodology

2.1 Selection of suitable AI-detectors

Firstly, it was necessary to select the most suitable tools with the ability to detect AI-generated content in Czech and Slovak languages. Hence, the Internet was searched on the 10th of October, for all publicly available tools that could detect such content. Publicly available tools found in this round of searching were, in turn, combined with the detectors that were researched in the initial study [9]. Overall, this approach yielded 22 AI detection tools: Compilatio, Duplichecker, Crossplag, GPT-2 output detector, Go winston, Gptzero.me, Zerogpt.com, Zerogpt.cc, ContentAtScale, Contentdetector, Copyleaks, Smodin.io, Plagiarismdetector, Scribbr, Undetectable.ai, Writer, CheckforAI, DetectGPT, OpenAI classifier, PlagiarismCheck, Writeful gpt detector and Sapling.ai.

To make our paper feasible and achieve meaningful results, we wanted to filter out low-quality tools. Therefore, we uploaded 4 AI-generated documents (2 in Czech, 2 in Slovak) to each tool. If the tool had correctly identified at least one, we would have included it in our paper. This approach was chosen because our main research question was detecting AI-generated text in Czech and Slovak documents. We decided to test the tools with multiple documents in both languages in case a lower accuracy of the tool could have caused the incorrect result.

The only tool that correctly recognised at least one of these texts as AI-generated was Compilatio, with a success rate of 50%. Four tools (Go winston, Copyleaks, PlagiarismCheck and Writeful gpt detector) did not recognise the document's language and thus refused to process it. Two tools (Undetectable.ai and Writer) had lagging web pages and could not provide any results. Three tools subject to testing in the previous study [9] (CheckforAI, DetectGPT, and OpenAI Classifier) were no longer operating. The rest of the tools incorrectly evaluated all four documents as human-written.

As we did not want to base our research on a single tool, we decided to include the GPT-2 Output Detector for comparison despite excluding it in the

previous step. We chose this tool because it was evaluated as the second-best publicly available tool in the initial study [9]. We did so because, according to the study, the best publicly available tool – Compilatio – was already part of our paper.

2.2 Test set

When creating the test set of the documents, we decided to unify as many parameters as possible to minimise the differences between the various tests across different languages. All human-written texts (in all languages – English, Czech and Slovak) were created by the paper’s first author. They were written within an upper limit of 500 characters, and all sentences were complete. It was essential to use documents that were not publicly available on the Internet and thus could not have been a part of the training set for ChatGPT (3.5 or 4) or one of the selected detectors.

Subsequently, the documents that were generated by ChatGPT had the same prompts in English, Czech and Slovak. ChatGPT generated all documents in the same language as the given prompt.

We used an online translation tool, DeepL [10], to translate Czech and Slovak documents into English. These documents were then used to test the AI detection tools to examine the effects of translation on the detectability of AI.

Overall, we had 15 categories. Each category consisted of 9 documents, so in total, 135 documents were subject to this paper. The categories were as follows: Written by human:

- in the Czech language
- in the Slovak language
- in the English language
- in Czech language and translated to English
- in Slovak language and translated to English

AI-generated:

- in the Czech language generated by ChatGPT3.5
- in the Slovak language generated by ChatGPT3.5
- in the English language generated by ChatGPT3.5
- in the Czech language generated by ChatGPT4
- in the Slovak language generated by ChatGPT4
- in the English language generated by ChatGPT4
- in Czech language generated by ChatGPT3.5 and translated to English
- in Slovak language generated by ChatGPT3.5 and translated to English
- in Czech language generated by ChatGPT4 and translated to English
- in Slovak language generated by ChatGPT4 and translated to English

2.3 Testing

The process of testing was done during two weeks. As the landscape of generative AI is evolving quickly [5,9], the period had to be as short as possible to ensure fair conditions for all tested tools. We uploaded every document one by one from the test set to both of the detectors and processed it. Consequently, the detector's score was recorded, and to ensure the integrity of the data, a screenshot of the result was taken.

Table 1: Division of documents based on the score given by an AI-detector.

Human-written documents (NEGATIVE):		
[100 - 80%) AI	False positive	FP
[80 - 60%) AI	Partially false positive	PFP
[60 - 40%) AI	Unclear	UNC
[40 - 20%) AI	Partially true negative	PTN
[20 - 0%] AI	True negative	TN
Documents generated by AI (POSITIVE):		
[100 - 80%) AI	True positive	TP
[80 - 60%) AI	Partially true positive	PTP
[60 - 40%) AI	Unclear	UNC
[40 - 20%) AI	Partially false negative	PFN
[20 - 0%] AI	False negative	FN

Both detectors provided a score on a scale from 0 – 100%, which indicated how confident they were that the document was AI-generated. To measure how correct the results from the detectors were, we divided them into ten categories according to Table 1, taken from [9].

We tested the detectors with a document set in the appropriate language and consisted of eighteen documents: nine human-written texts + nine ChatGPT-generated documents. In one case, we compared the human-written documents to those generated by ChatGPT3.5, and in the other case, we compared the same human-written documents to those generated by ChatGPT4.

When testing the detectors with the translated texts, we used the same human-written texts in Czech and Slovak and AI-generated texts in Czech and Slovak as in the previous tests. This time, all AI-generated and human-written documents were translated to English through DeepL to be processed by GPT-2 Output Detector, which only worked with English.

2.4 Relevant metrics

We decided to use accuracy, sensitivity and specificity as the relevant metrics for the paper. In this paper, we counted accuracy as `accuracy_semibin` as defined by Weber-Wulff et al. [9]. Sensitivity and specificity are commonly used for evaluating the efficiency of classifying tests [11].

Accuracy shows how likely the detector is to make the correct decision. Partially true decisions reward accuracy with half a point; other results count as incorrect.

$$Accuracy = \frac{TP + TN + 0.5 * (PTP + PTN)}{n \text{ of all documents}} * 100$$

Sensitivity is the probability that the detector correctly detects AI-generated content among the AI-generated documents.

$$Sensitivity_{original} = \frac{TP}{TP + FN} * 100$$

Subsequently, we made a slight change to the original formula of sensitivity. Since the results from the detectors were of ten categories, and this formula is designed for binary classification, it would not account for partial results. So, we decided to reward partially correct results with a lower weight, and in the denominator of the formula, we included the number of all AI-generated documents.

$$Sensitivity = \frac{TP + 0.5 * PTP}{n \text{ of AI-generated documents}} * 100$$

Specificity is the probability that the detector correctly evaluates human-written text as human-written. High specificity minimises the portion of falsely labelled human-written texts as AI-generated.

$$Specificity_{original} = \frac{TN}{TN + FP} * 100$$

As with sensitivity, we updated the original formula of specificity. We did this to reward partially correct human classifications and include the number of all human-written documents in the denominator. Specificity for this paper was computed with this formula.

$$Specificity = \frac{TN + 0.5 * PTN}{n \text{ of human-written documents}} * 100$$

3 Results

The results of this paper interestingly show that the accuracy of both tools, Compilatio and GPT-2 Output detector, significantly dropped compared to the results from the initial research [9]. English texts generated by ChatGPT4 were not recognised by either of the tools, and all passed as human-written.

The results of our testing revealed that the GPT-2 Output detector was incapable of correctly classifying Czech and Slovak documents and classified all of them as human-written. Compilatio, on the other hand, could process both Czech and Slovak documents. It was more accurate (67%) with Slovak

documents than Czech documents (56%). Furthermore, it was more likely to detect AI-generated documents while keeping its accuracy slightly lower than English texts.

In almost all categories, we could see that text generated by ChatGPT4 was less detectable. In the one case where Compilatio seemed more accurate and sensitive towards detecting Czech content from ChatGPT4, there was an insignificant difference of one correctly classified document.

Overall, the tools were unreliable in delivering correct answers and thus their judgement should be taken cautiously. Nevertheless, Compilatio performed well with the English texts, managing to have no false positives while still being able to detect ChatGPT3.5.

Table 2: Results for ChatGPT3.5.

GPT3.5	English		Czech		Slovak	
	Compilatio	GPT-2 O.D.	Compilatio	GPT-2 O.D.	Compilatio	GPT-2 O.D.
Specificity	100%	60%	56%	100%	61%	100%
Sensitivity	22%	28%	56%	0%	72%	0%
Accuracy	61%	45%	56%	50%	67%	50%

Table 3: Results for ChatGPT4.

GPT4	English		Czech		Slovak	
	Compilatio	GPT-2 O.D.	Compilatio	GPT-2 O.D.	Compilatio	GPT-2 O.D.
Specificity	100%	60%	56%	100%	61%	100%
Sensitivity	0%	0%	67%	0%	67%	0%
Accuracy	50%	32%	61%	50%	64%	50%

3.1 Effects of translation

In the following test, we sought to examine how useful the detectors were when presented with translated documents. Compilatio, when presented with translated texts in Czech, performed with higher specificity (78% compared to 56%) and a bit lower accuracy (44% compared to 56%) but a much lower sensitivity (56% compared to 11%); hence its ability to detect AI-generated text was considerably worse. The same applied to documents in the Slovak language; here, the sensitivity dropped to the bare minimum (0% for ChatGPT3.5 and 11% for ChatGPT 4); hence, the detector was nearly unable to detect AI-generated text.

GPT-2 Output Detector was a very different case; here, we had a tool that could not operate on Czech and Slovak documents and thus had zero sensitivity. How-

Table 4: Results for Compilatio.

Compilatio	Czech Documents				Slovak Documents			
	Original		Translated to EN		Original		Translated to EN	
	GPT3.5	GPT4	GPT3.5	GPT4	GPT3.5	GPT4	GPT3.5	GPT4
Specificity	56%	56%	78%	78%	61%	61%	89%	89%
Sensitivity	56%	67%	11%	22%	72%	67%	0%	11%
Accuracy	56%	61%	44%	50%	67%	64%	47%	50%

ever, when the texts were translated into English, the tool’s performance dramatically increased. When compared to Compilatio, it had a little higher specificity (61% compared to 56%), lower sensitivity (33% compared to 56%) and a bit lower accuracy (47% compared to 56%) with translated Czech documents. With Slovak documents, it also performed surprisingly well; it had a decent specificity (83%), so it did not generate too many false positives and it had a notable sensitivity (44%) and accuracy (64%). Nevertheless, it was less decisive than Compilatio and more often it gave partial or unclear results. Compilatio made definitive results (TP, TN, FP, FN) in 96% cases, compared to GPT-2 Output Detector’s 75

Table 5: Results for GTP-2 Output Detector.

GPT-2 O.D.	Czech Documents				Slovak Documents			
	Original		Translated to EN		Original		Translated to EN	
	GPT3.5	GPT4	GPT3.5	GPT4	GPT3.5	GPT4	GPT3.5	GPT4
Specificity	100%	100%	61%	61%	100%	100%	83%	83%
Sensitivity	0%	0%	33%	33%	0%	0%	44%	39%
Accuracy	50%	50%	47%	47%	50%	50%	64%	61%

4 Discussion

At the time of writing of this paper, there was only one publicly available AI detector that was able to detect AI-generated content in the Czech and Slovak languages: Compilatio. Its performance with Czech and Slovak documents was not much better than deciding by flipping a coin. With both languages, the detector had nearly 60% specificity, the rest were false positives.

Our findings showcased that the accuracy with English documents and ChatGPT3.5 text of both Compilatio and GPT-2 Output Detector dropped from April 2023 when we conducted the initial study [9] till the conducting of this paper in October 2023. Compilatio went from 77% to 61%, and GPT-2 Output Detector dropped from 73% to 45%. Nevertheless, with English documents, Compilatio had a specificity of 100% and therefore had no false positives.

ChatGPT4 showed that it could generate English content that was less likely to be detected. Neither of the tools detected English documents from ChatGPT4. Interestingly, both tools detected the Slovak and Czech content from ChatGPT4 just as likely as the content from ChatGPT3.5. It showed that the premium version of ChatGPT could only generate better content in English.

5 Conclusions

All in all, this paper demonstrated that the current state of detection of AI-generated content in Czech and Slovak languages does not deliver satisfying results. It is thus advisable to avoid relying solely upon the results provided by such tools. Translating the documents to English and then uploading them to the detectors allowed us to use an English-only tool, GPT-2 Output Detector and get comparable results to Compilatio. This could imply that translation tools preserve the human characteristics of human-written text. Still, further research has to be done to confirm the actual reasons for such an outcome. Ultimately, this paper demonstrated that English texts generated by ChatGPT4 are generally less detectable than those generated by ChatGPT3.5. Such an outcome hints towards the rapid progress in AI-generated content, which in many regards remains faster than any efforts and tools targeted at AI detection.

References

1. Casal, J. & Kessler, M. Can linguists distinguish between ChatGPT/AI and human writing?: A study of research ethics and academic publishing. *Research Methods In Applied Linguistics*. **2**, 100068 (2023)
2. Tang, R., Chuang, Y. & Hu, X. The science of detecting llm-generated texts. *ArXiv Preprint ArXiv:2303.07205*. (2023)
3. Compilatio Anti-plagiarism Software | Plagiarism Prevention and Detection, <https://www.compilatio.net/en>. Last accessed 10 Oct 2023
4. GPT-2 Output Detector, <https://openai-openai-detector.hf.space/>. Last accessed 10 Oct 2023
5. Foltýnek, T., Bjelobaba, S., Glendinning, I., Khan, Z. R., Santos, R., Pavletic, P., Kravjar, J.: ENAI Recommendations on the ethical use of Artificial Intelligence in Education. *International Journal for Educational Integrity* **19**(1), 12 (2023)
6. De Angelis, L., Baglivo, F., Arzilli, G., Privitera, G., Ferragina, P., Tozzi, A. & Rizzo, C. ChatGPT and the rise of large language models: the new AI-driven infodemic threat in public health. *Frontiers In Public Health*. **11** pp. 1166120 (2023)
7. Chaka, C. Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools. *Journal Of Applied Learning And Teaching*. **6** (2023)
8. Elkhataf, A., Elsaid, K. & Almeer, S. Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *International Journal For Educational Integrity*. **19**, 17 (2023)
9. Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltýnek, T., Guerrero-Dib, J., Popoola, O., Šigut, P. & Waddington, L. Testing of detection tools for AI-generated text. *ArXiv Preprint ArXiv:2306.15666*. (2023)

10. DeepL Translate: The world's most accurate translator — [deepl.com](https://www.deepl.com/translator), <https://www.deepl.com/translator>. Last accessed 20 Oct 2023
11. Parikh, R., Mathai, A., Parikh, S., Sekhar, G. & Thomas, R. Understanding and using sensitivity, specificity and predictive values. *Indian Journal Of Ophthalmology*. **56**, 45 (2008)

Part II

Evaluation Methods

Does Size Matter?

Comparing Evaluation Dataset Size for the Bilingual Lexicon Induction

Michaela Denisová and Pavel Rychlý

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
`{449884,pary}@mail.muni.cz`

Abstract. Cross-lingual word embeddings have been a popular approach for inducing bilingual lexicons. However, the evaluation of this task varies from paper to paper, and gold standard dictionaries used for the evaluation are frequently criticised for occurring mistakes. Although there have been efforts to unify the evaluation and gold standard dictionaries, we propose a new property that should be considered when compiling an evaluation dataset: size. In this paper, we evaluate three baseline models on three diverse language pairs (Estonian-Slovak, Czech-Slovak, English-Korean) and experiment with evaluation datasets of various sizes: 200, 500, 1.5K, and 3K source words. Moreover, we compare the results with manual error analysis. In this experiment, we show whether the size of an evaluation dataset impacts the results and how to select the ideal evaluation dataset size. We make our code and datasets publicly available ¹

Keywords: Cross-lingual word embeddings, bilingual lexicon induction, evaluation dataset’s size.

1 Introduction

Cross-lingual word embeddings (CWEs) have drawn attraction among researchers due to their ability to connect meanings across languages. CWEs enable the alignment of two (or more) sets of independently trained monolingual word embeddings (MWEs) into one shared cross-lingual space, where similar words obtain similar vectors [18].

Given this property, they have proven useful in many NLP applications, for instance, document classification [16], machine translation [4,6] or language learning [1].

A broadly used way to evaluate these models is through the bilingual lexicon induction (BLI) task. In this task, the objective is to find top k target words for a source word whose word vectors are the closest in the aligned vector space. This is achieved typically by computing cosine similarity between the

¹ https://github.com/x-mia/Eval_set_size

source and target word vectors. Then, the retrieved word pairs are compared to those occurring in the evaluation dataset, often referred to as gold-standard dictionary [18].

However, the evaluation process and evaluation datasets are not unified, and they vary from paper to paper, using various training parameters, evaluation criteria, and evaluation datasets [17,14,3]. This obstructs our ability to accurately assess the results, monitor any progress in new models, and compare models with each other.

Moreover, the most popular evaluation datasets MUSE [7] are often criticised since they were compiled automatically and contain much noise in the form of occurring mistakes, such as inflected word forms (*wave*, singular - *Wellen*, plural, German dataset), a different part of speech (*darkness*, noun - *temné*, dark, adjective, Slovak dataset), same word translations (*android* - *android*, Korean dataset) [9]. They often have disproportional part-of-speech (POS) distribution, where a quarter of data consists of proper nouns that do not carry any lexical meaning and cannot indicate the performance adequately. For example, *Barack Obama*, *Skype*, *Bruno*, *Wisconsin*, etc. [15].

Some efforts have focused on uniting the evaluation by investigating how different training parameters influence the results [10]. Furthermore, some studies suggest consolidating the evaluation datasets through equal POS representations [15,13]. Nonetheless, other factors and properties of the evaluation dataset should also be considered.

One of them and one of the unifying steps is determining the size of the evaluation dataset. MUSE evaluation datasets contain 1.5K source words, which have become a standard for the BLI.

In this paper, we investigate whether the number of source words in the evaluation dataset impacts the results. We explore how many source words are enough to assess the quality of the model. Our motivation is to study whether we can use fewer source words to create a high-quality evaluation dataset that reflects the model's performance precisely while minimising the time and effort of the human annotators to compile it.

We evaluate three popular baseline CWE models, i.e., MUSE [7], VECMAP [2,3], RCLS [14] on three diverse language pairs: distant language pair (Estonian-Slovak), close language pair (Czech-Slovak), and language pair that do not share a script (English-Korean). We utilise evaluation datasets of different sizes: 200, 500, 1.5K, and 3K source words, and observe how the results change. We compare the results against human performance to ensure the precise reflection of the resulting quality.

Our contribution is manifold:

- We provide an evaluation of three common baseline models with evaluation datasets of various sizes.
- We set the appropriate number of source words for the efficient, high-quality evaluation dataset that is less time-consuming to compile and indicate accurate results.

- We propose another unifying property for evaluation datasets to make the evaluation process comparable and reproducible for other researchers.

Our paper is structured as follows. In Section 2, we present the details of baselines and training, our datasets and the metrics used. In Section 3, we evaluate the models with datasets of various sizes, show the outcomes and discuss the results. Finally, we offer concluding remarks in Section 4.

2 Bilingual Lexicon Induction

The BLI task includes several aspects, such as evaluation datasets used, evaluation metrics, and selected baselines and training. We introduce them in this section.

Evaluation Datasets. Since we wanted to assess only the impact of the size, the aim was to make each size group of source words as similar as possible.

The Estonian-Slovak (et-sk) evaluation dataset was compiled using the Estonian-Slovak dictionary from Denisová (2021) [8]. This dataset claims 40% accuracy; therefore, we post-processed the word pairs manually after selection. We randomly sampled 3K source words and then randomly split them into 200, 500, and 1.5K source words for the subsequent evaluation.

The evaluation dataset for Czech-Slovak (cs-sk) was constructed manually mostly from words that are different in both languages (e.g., *želva* - *korytnačka*, *turtle*). We applied the same procedure as for the Estonian-Slovak evaluation dataset, i.e., we compiled a 3K source-word dataset and randomly sampled 1.5K, 500, and 200 source words.

For English-Korean (en-ko), we used the open-source evaluation dataset MUSE, which consists of 1.5 source words (English-Korean test set). Afterwards, we randomly selected 500 and 200 source words for the subsequent evaluation. To extend this dataset, we randomly sampled another 1.5K source words from the full English-Korean MUSE dataset. Afterwards, we combined them with the MUSE evaluation dataset to create a 3K-source-words dataset.

Metrics. The most common reported metric in the BLI task is precision (%). Precision or $P@k$ is the ratio of True Positives (TP) to the sum of the True Positives and False Positives (FP) defined by the following formula:

$$P = TP / (TP + FP)$$

Where k represents the number of top target words retrieved for a source word. In this paper, we compute $P@1$, i.e., we retrieve one closest target word for each source word.

Baselines. MUSE [7] is a generative-adversarial-network-based model in the unsupervised setting (MUSE-U). The supervised (MUSE-S) setting and setting that relies on identical strings (MUSE-I) uses iterative Procrustes alignment.

VEC_{MAP} is a framework that encompasses various stages, including orthogonal mapping, re-weighting, and dimensionality reduction, within its supervised settings (VM-S, VM-I) [2]. In its unsupervised setting (VM-U) [3], VEC_{MAP} employs a robust iterative self-learning procedure.

RCLS is an orthogonal-mapping-based method with implemented convex relaxation in the retrieval stage. We trained this method in the supervised setting only.

Seed Lexicons. The Seed lexicons used in supervised training were compiled the same way as evaluation datasets. For Estonian-Slovak, we randomly sampled 5K source words from Denisová (2021)’s dataset [8]. For Czech-Slovak, we automatically constructed a 5K-source-words dataset consisting of identical words from the MWE vocabularies. For English-Korean, we used MUSE training dataset [7].

Training. The default settings closely adhere to the training outlined in [7] for the MUSE model, and VM-S and VM-I are presented in [2]. The parameters for VM-U follow the training procedures from [3]. Additionally, RCLS training settings align with those described in [14].

During the training, we experimented with two MWEs. We used pre-trained FastText embeddings [11] for Estonian, Slovak, Czech, English, and Korean, which were trained on texts from Wikipedia² with dimension 300.

The second pre-trained embeddings were provided by SketchEngine [12].³ These embeddings were trained with the same method [5] but on different data (web corpora), with dimensions 100 for Estonian-Slovak and English-Korean, and 300 for Czech-Slovak.

3 Evaluation

In the evaluation process, we assessed all three models on Estonian-Slovak, Czech-Slovak, and English-Korean with the split datasets into four groups: 200, 500, 1.5K, and 3K source words. We extracted one target word for each source word by computing the cosine similarity between the source and target word vector. Then, we calculated P@1. Tables 1, 2, and 3 show the results.

Tables 1 and 2 show that the difference between the precision for both groups fluctuates wildly within a margin of approximately 15%. The best results were achieved for the Estonian and Czech in combination with Slovak when the 3K-source-word datasets were used.

This could mean that we get more precise results with datasets containing more source words or that the underlying distribution varies significantly after splitting the dataset.

The exemption was Table 3, the English-Korean language pair. In the majority of cases, the best results were gained with the 1.5K-source-word dataset,

² <https://www.wikipedia.org/>

³ <https://embeddings.sketchengine.eu/>

Table 1: The results for the Estonian-Slovak language combination.

et-sk (%)	FastText				SketchEngine			
	200	500	1.5K	3K	200	500	1.5K	3K
MUSE-S	17.34	18.93	21.37	23.18	26.53	27.02	32.30	36.14
MUSE-I	10.20	13.40	15.30	16.65	27.55	24.68	28.82	32.03
MUSE-U	11.73	12.76	13.52	15.64	20.40	20.85	23.80	27.14
VM-S	19.89	25.53	26.88	30.72	28.57	28.72	34.81	38.85
VM-I	17.34	18.29	22.18	24.60	21.93	22.76	26.63	30.15
VM-U	15.30	16.17	19.67	21.72	22.95	22.12	26.63	29.80
RCLS	16.83	19.78	22.99	27.05	27.55	26.59	34.73	38.28

Table 2: The results for the Czech-Slovak language combination.

cs-sk (%)	FastText				SketchEngine			
	200	500	1.5K	3K	200	500	1.5K	3K
MUSE-S	58.08	62.10	64.99	68.72	62.26	65.89	71.50	75.72
MUSE-I	59.59	61.68	64.92	68.93	61.61	65.68	70.97	75.48
MUSE-U	60.60	62.31	65.51	69.25	61.00	65.68	70.97	75.44
VM-S	59.09	60.63	66.41	69.13	62.62	65.47	71.50	75.84
VM-I	59.09	64.42	68.66	72.10	61.61	65.89	71.42	75.52
VM-U	59.09	64.21	68.58	72.10	61.61	65.89	71.50	75.60
RCLS	57.57	61.05	64.32	68.04	64.14	67.36	72.70	76.48

Table 3: The results for the English-Korean language combination.

en-ko (%)	FastText				SketchEngine			
	200	500	1.5K	3K	200	500	1.5K	3K
MUSE-S	13.91	13.57	17.44	15.91	16.49	19.82	21.23	19.00
MUSE-I	11.34	14.22	17.16	15.80	10.30	15.51	14.64	13.90
MUSE-U	10.30	11.42	13.94	12.78	12.37	13.36	12.05	11.63
VM-S	29.38	29.52	35.31	33.80	21.13	20.90	23.75	21.58
VM-I	20.61	17.67	21.72	19.03	13.91	15.30	15.41	13.43
VM-U	12.37	14.22	16.53	14.51	6.70	5.81	6.51	5.63
RCLS	30.92	27.80	34.40	32.54	21.13	20.90	22.91	20.25

within a margin of approximately 5%, which is not as distinct as in the previous two groups. This dataset came from a different source than the other ones, and as the only one, it was compiled automatically, which might be the reason for various outcomes.

Moreover, when comparing the results with different MWEs, the models trained with SketchEngine MWEs outperformed the models trained with Fast-Text MWEs in most cases. We explain the differences in Subsection 3.1 in further detail.

In the next step, we split all three 3K-source-word datasets into six random groups of 500 source words. Afterwards, we evaluated VM-S with these datasets as an example. The objective was to observe whether the large gaps would be preserved in a different setup or whether the changed word distribution would bring more balance. Table 4 shows the development of this experiment.

Table 4: The results of 3K-headword evaluation datasets split into groups of 500.

VM-S	ET-SK		CS-SK		EN-KO	
	FastText	SketchEngine	FastText	SketchEngine	FastText	SketchEngine
I.	26.73	29.34	64.64	70.50	28.33	17.45
II.	21.42	25.10	61.89	68.00	28.45	17.78
III.	26.30	33.04	60.45	67.28	31.12	18.67
IV.	22.82	30.65	58.10	63.36	27.55	20.00
V.	22.73	30.31	57.74	64.22	29.35	19.07
VI.	21.61	29.55	53.52	57.64	30.06	18.88

Given Table 4, the gaps for each group of evaluation source words were reduced, remaining within the margin of approximately 8%. This suggests that random sampling seemingly might preserve the underlying distribution; however, the variance in the real scenario is more significant.

3.1 Error Analysis

Due to the inconsistencies in the outcomes, we performed manual error analysis for the Estonian-Slovak and Czech-Slovak language pairs while using the model VM-S as an example. Table 5 outlines the results.

Based on the results stated in Table 5, we can observe that the gaps reduced significantly, staying within the margin up to 4%. The best result was achieved twice with the 1.5K-source-word datasets, once with the 200- and 500-source-word datasets.

The reasons behind the large gaps between the results were twofold. Firstly, the top first target word that the model found was not in the evaluation dataset, although it was correct, e.g., *ajajärk* (time period, era, epoch) - *obdobie* (VM-S), *doba* (evaluation dataset).

Table 5: Manual error analysis of the results of the model VM-S for Estonian-Slovak and Czech-Slovak.

VM-S	ET-SK		CS-SK	
	FastText	SketchEngine	FastText	SketchEngine
3K	45.41	56.51	86.57	94.41
1.5K	47.61	59.67	87.28	94.83
500	46.59	58.51	86.31	94.94
200	46.93	60.71	86.86	94.44

This happened quite often because we did not include such a target word in the evaluation dataset, or the source words with multiple target words were randomly spread out in different datasets during the splitting. For example, the Estonian source word *puhuma* (*to blow*) had multiple target words such as, *pofúkať*, *fúkať*, *trúbiť*, *vanúť*, *zaviať*, *viať*, and in the 200-source-word dataset got the target word *zaviať* that was not the top one (which was *fúkať*) but the top second or third target word that the model found.

The second common reason was the uneven distribution of out-of-the-vocabulary (OOV) words. These were words that were not in the MWEs, low-frequency words, and words left out during the training. For example, *řeřicha* (*garden cress*), *pulec* (*tadpole*), *drobek* (*crumb*), *segisti* (*faucet*), *ahing* (*fish-spear*), etc.

On top of that, we analysed the gaps between SketchEngine and FastText MWEs. A closer look revealed that models trained with FastText MWEs were more likely to find a correct equivalent for proper nouns (e.g., *Clara*, *Emma*, *Erik*, *Phillip*, etc., see Table 6, type A) which have a bigger representation in the English-Korean datasets than in Czech-Slovak or Estonian-Slovak evaluation datasets.

Moreover, the English-Korean dataset contained a lot of noise in the form of words translated with the same word (e.g., *vms-vms*, *pgm-pgm*, etc., see Table 6, type B), for which the models trained with FastText were more likely to find a target word from the evaluation dataset.

On the other hand, models trained with SketchEngine MWEs were better at finding more accurate translation equivalents rather than words with lexical-semantic meanings (e.g., *hnědá*), see Table 6, type C). Additionally, they outperformed the FastText MWEs on the vocabulary, slang (e.g., *emps*) and low-frequency words (e.g., *stýskat*, *chasník*, etc., see Table 6, type D). The examples are displayed in Table 6

4 Conclusion

In this paper, we have investigated whether the standard 1.5K source words used in the evaluation datasets are enough to assess the CWE model accurately

Table 6: The differences between the models trained with FastText and SketchEngine MWEs (examples from EN-KO, ET-SK, and CS-SK trained with the VM-S model). (FT = FastText; SkeEng = SketchEngine)

Type	SRC	ED	FT	SkeEng	Description
A	Clara	클라라	클라라	외가에서	proper names
	Emma	엠마	엠마	희진	
	Erik	에릭	에릭	동료인	
	Phillip	필립	필립	웅은	
B	vms	vms	vms	램도	same word with same word
	pgm	pgm	pgm	변환하고	
C	hnědá	hnedá (brown)	žltohnedá	hnedá	precise translations
D	stýskat	cnief (to miss)	-	cnief	low-frequency words, slang
	chasník	mládenec (young man)	-	chasník	
	emps	mamka	-	mamka	

or whether we need more or fewer source words. We have experimented with evaluation datasets of various sizes: 200, 500, 1.5K, and 3K source words. Furthermore, we have split the 3K-source-words evaluation dataset into six random groups of 500 to observe how the outcomes would change.

Afterwards, we provided a manual error analysis, focusing on gaps between the results with different evaluation datasets. And we analysed the differences between the models trained with FastText and SketchEngine MWEs. We explained them and presented examples.

In conclusion, when splitting the datasets randomly, the results oscillated intensely within a margin of up to 15%. However, the manual error analysis revealed that the actual results faintly varied between 1-4%, remaining approximately the same within all datasets.

This outcome suggests that the random splitting of datasets does not ensure an equal underlying distribution within all the datasets. Moreover, it shows that the result strongly depends on the appropriate vocabulary choice rather than on the size of the dataset. This confirms the exemptional results from the English-Korean language pair evaluated on a dataset from a different resource than the others.

Generally, when selecting the size of the evaluation dataset, comparable results could be achieved even with a smaller dataset when the focus is on the quality of the chosen vocabulary for the evaluation dataset.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Akyurek, E., Andreas, J.: Lexicon learning for few shot sequence modeling. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4934–4946. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.acl-long.382>
2. Artetxe, M., Labaka, G., Agirre, E.: Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. pp. 5012–5019 (2018). <https://doi.org/10.1609/aaai.v32i1.11992>
3. Artetxe, M., Labaka, G., Agirre, E.: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 789–798. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/P18-1073>
4. Artetxe, M., Labaka, G., Agirre, E.: Unsupervised statistical machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3632–3642. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/D18-1399>
5. 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). <https://doi.org/10.48550/arXiv.1607.04606>
6. Chronopoulou, A., Stojanovski, D., Fraser, A.: Improving the lexical ability of pre-trained language models for unsupervised neural machine translation. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 173–180. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.naacl-main.16>
7. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. ArXiv **abs/1710.04087** (2017). <https://doi.org/10.48550/arXiv.1710.04087>
8. Denisová, M.: Compiling an Estonian-Slovak dictionary with English as a binder. In: Proceedings of the eLex 2021 conference. pp. 107–120. Lexical Computing CZ, s.r.o. (2021), https://ellex.link/ellex2021/wp-content/uploads/2021/08/eLex_2021_06_pp107-120.pdf
9. Denisová, M., Rychlý, P.: When word pairs matter: Analysis of the english-slovak evaluation dataset. In: Recent Advances in Slavonic Natural Language Processing (RASLAN 2021). pp. 141–149. Brno: Tribun EU (2021), <https://nlp.fi.muni.cz/raslan/2021/paper3.pdf>
10. Glavaš, G., Litschko, R., Ruder, S., Vulić, I.: How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 710–721. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/P19-1070>

11. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018), <https://aclanthology.org/L18-1550>
12. Herman, O.: Precomputed word embeddings for 15+ languages. *RASLAN 2021 Recent Advances in Slavonic Natural Language Processing* pp. 41–46 (2021), <https://www.sketchengine.eu/wp-content/uploads/2021-Precomputed-Word-Embeddings.pdf>
13. Izbicki, M.: Aligning word vectors on low-resource languages with Wiktionary. In: *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*. pp. 107–117. Association for Computational Linguistics (2022), <https://aclanthology.org/2015.mtsummit-papers.27>
14. Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., Grave, E.: Loss in translation: Learning bilingual word mapping with a retrieval criterion. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pp. 2979–2984. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/D18-1330>
15. Kementchedjhieva, Y., Hartmann, M., Søgaard, A.: Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pp. 3336–3341. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/D19-1328>
16. Klementiev, A., Titov, I., Bhattarai, B.: Inducing crosslingual distributed representations of words. In: *International Conference on Computational Linguistics*. pp. 1459–1474 (2012), <https://aclanthology.org/C12-1089>
17. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. *ArXiv abs/1309.4168* (2013). <https://doi.org/10.48550/arXiv.1309.4168>
18. Ruder, S., Vulić, I., Søgaard, A.: A survey of cross-lingual word embedding models. *The Journal of Artificial Intelligence Research* **65**, 569–631 (2019). <https://doi.org/10.48550/arXiv.1706.04902>

Reproducibility and Robustness of Authorship Identification Approaches

Adam Karásek and Zuzana Nevěřilová

Natural Language Processing Centre
Faculty of Informatics
Botanická 68a, Brno, Czech Republic

Abstract. Authorship identification, framed as a classification task, assigns a digital text to a known author. State-of-the-art algorithms for this task often lack evaluation across diverse datasets. This paper re-implements and evaluates three approaches on three different datasets, exploring the robustness of algorithms on various text types (e.g., emails, articles, instant messages).

Not all the published methods are fully reproducible. However, reasonable parameters were selected if they were not part of the original paper. The evaluation of the ensemble model shows it is somewhat robust on different texts and different numbers of potential authors.

Keywords: authorship identification, evaluation, reproducibility

1 Introduction

Authorship identification is a classification task that assigns a human-written digital text to an author from a known set of authors. There are many different state-of-the-art algorithms for classifying authors of text based on numerous classification algorithms and text processing techniques. However, papers proposing these solutions often provide their evaluation only on one selected dataset. The research question is how robust the distinct algorithms on datasets of different types of text (e.g., emails, articles, or instant messages) are.

In this paper, we re-implemented three different approaches and evaluated them on three different datasets. Apart from robustness, we examined the reproducibility of the published papers. In Section 2, we describe the selected approaches to authorship identification. Section 3 describes the datasets we selected for the evaluation. We aimed to select heterogeneous data in English. In Section 4, we describe our re-implementation. Section 5 describes the evaluation of the count vector ensemble model, and Section 6 draws conclusions about the robustness of the model.

2 Related Work

Authorship identification, also called authorship attribution in some literature[9], is part of a broader field of authorship analysis. There are two other tasks, as

stated in [10]. Authorship verification is a mechanism for deciding whether a specific individual wrote an anonymous text. Authorship characterization presumes the author's characteristics, such as gender, age, social background, etc.

Traditionally, identifying the author of an anonymous text was done using stylometric features. Over the years, over 1000 stylometric features of different types, such as lexical, syntactic, structural, content-specific, and idiosyncratic. Nonetheless, there is no consensus on which features or set of features are most helpful in identifying the author of a given text. Different stylometric features can be suitable based on the type and properties of the examined text. Measured features are, for instance, average word length, punctuation rate, occurrence of special characters, etc. [1]

2.1 Count Vector Ensemble Model

The first model we selected for our experiment was introduced by [2] using a count vector for feature extraction and an ensemble of three classification models as a classifier. The count vector calculates the frequency of each word, called *term frequency*, of the input text. This approach measures how many times an author uses specific words. Therefore, the classification model can recognize the author based on their use of words. The authors of [2] used random forest, extreme gradient boosting (xgboost, XGB), and multilayer perception (MLP) as a classification model in the ensemble. With this setup, they have reached 97 % accuracy on 10 authors and 79 % accuracy on 20 authors on a news articles dataset. The dataset was composed from over 140 000 news articles from 15 american news websites.

2.2 Email Detective

As a second model for the experiment, we chose a neural network proposed by [7] with two inputs. The first input processes text using the *word2vec* method for text embedding. Word2vec transforms words into high-dimensional vectors. These vectors reflect the words' meaning so that semantically similar words are close to each other in the vector space [8].

Unconventionally, the authors of Email Detective use the same method to embed characters (excluding spaces), not words. Next, the embedded characters were input into a BiLSTM layer. After processing the text input, it was concatenated with 10 stylometric values gathered from the email header. The full set of text features is afterward classified with a dense layer.

The authors of [7] used the Enron dataset to evaluate their model. Email Detective achieved an accuracy of 98.9 % for 10, 92.9 % for 25, and 89.5 % for 50 authors.

2.3 BertAA

The third implemented algorithm is a transformer-based classification neural network introduced by [6]. Their model consists of a pre-trained BERT fine-tuned on an authorship attribution dataset with a dense layer for classification.

They used Enron emails, IMDb Authorship Attribution, and Blog Authorship Attribution corpus for evaluating the model. For the Enron dataset, the model achieved an accuracy of 99.95 % for 5 authors, 99.1 % for 10 authors, and 98.7 % for 25 authors. The IMDb dataset's accuracy reached 99.6 % for 5 authors, 98.1 % for 10 authors, and 93.2 % for 25 authors. Furthermore, for the Blog dataset, the model attained an accuracy of 61.3 % for 5 authors, 65.4 % for 10 authors, and 65.3 % for 25 authors.

3 Datasets

We selected datasets where authorship is part of the annotation. At the same time, we wanted the evaluation data to be as diverse as possible. We therefore selected emails, social media posts, and news texts.

3.1 Enron Emails

The Enron emails dataset is a publicly available dataset of emails from about 150 authors published by the Federal Energy Regulatory Commission as part of an investigation of Enron Corporation. We used a version of the dataset partially preprocessed by the CMU School of Computer Science. The dataset contains about 500,000 emails from the management of Enron. [5]. All of the emails were sent between 1997 and 2002. [11]

As part of gathering all the email texts into one CSV¹ file, we separated the email body written by the author from the rest of the email file. The start of the email body is deterministic and, therefore, easy to find. The last line of the email header always starts with "X-FileName: ", and the next line is already the email body. The end of the email body written by the authors was harder to find. Often, email file contains forwarded messages. This occurs in the dataset in two ways. The first is standardized, probably done by an email client, in which the other author's text starts with one of the following:

- "---- Forwarded by"
- "---- Original Message"
- "---- Original Appointment"

Therefore, we sliced away the part of an email body starting with the phrases listed above.

The second form of another author's text is also resending someone else's email, but probably due to copying and pasting without the exact structure. We noticed that these parts of text often contain phrases such as "To: ", "From: ", and "Send by: ". So, we cut out the part starting with these phrases. We also removed the signatures. The texts in the dataset have an average of 400 to 500 characters.

¹ Comma-separated values

3.2 Techcrunch Articles

The second dataset comprises articles from `techcrunch.com`, an online newspaper focused primarily on startups and tech companies. We obtained the dataset on Kaggle [4]. There was no need to preprocess the dataset other than deleting all unnecessary columns so that we kept only the author and text columns. The texts in the dataset have an average of 3000 to 3500 characters.

3.3 Telegram Messages

The last dataset is gathered from the biggest Telegram group focused on cryptocurrencies. Data were published by Kaggle user Anton [3]. We chose the “OKEx official group” as the biggest of the five datasets in Telegram. The dataset was published in JSON format, but we transformed it into a CSV file for easier processing and standardization with other datasets and deleted all redundant information, keeping only the author and text. This dataset is characterized by frequent usage of emojis. No other preprocessing was needed. The texts in the dataset have an average of 200 to 220 characters.

3.4 Training and Evaluation Subsets

We constrained all datasets in the following way:

1. The task complexity increases with the number of potential authors. We, therefore, selected k authors with the largest number of documents.
2. Every document from the document set has to be at least 100 characters long. This constraint removes documents (e.g., emails and instant messages) containing only one sentence (such as “I’ll be there.”) for which it is impossible to assign an author.
3. Since the dataset should be balanced, we select l random documents written by each selected author.

We created three experiment sets for each dataset with $k \in \{5, 10, 25\}$. Parameter l is different for each dataset and parameter k . This is done to ensure the maximal size of experiment sets while keeping them balanced. The number of documents per author is shown in the table below.

Table 1: Number of documents per author in experiment set

Dataset	$k = 5$	$k = 10$	$k = 25$
Enron	$l = 4000$	$l = 2000$	$l = 800$
Telegram	$l = 1000$	$l = 650$	$l = 470$
Techcrunch	$l = 2500$	$l = 1200$	$l = 250$

4 Algorithm implementation

We compare three algorithms because they are relatively recent and reported a high accuracy rate. The re-implementation in Python is available at GitHub².

4.1 Count Vector Ensemble

We implemented the count vector ensemble model described in [2]. We created a class to encapsulate the xgboost classifier, multilayer perceptron, and random forest classifier. We separated 10 % of the learning dataset into a validation dataset and used the rest as a training dataset. Then, we transformed the text into a count vector.

[2] did not specify the number of layers and nodes in the MLP classifier. Therefore, we experimented with different settings to attain the best results. We implemented the MLP with three hidden dense layers with 4096, 2048, and 1024 nodes, respectively, and a ReLU activation in a forward direction with a dropout layer set to 0.5 after each hidden layer, including L2 regularization in each dense layer. The count vector size determined the input layer dimension. The output layer is a softmax with the number of nodes = number of authors.

The output shape is a one-hot encoding for the MLP model and a one-dimensional token for random forest and xgboost classifier.

The rest of the hyperparameters were described in the original paper; therefore, we used the values provided by [2]. We set the loss function to categorical cross-entropy, and we used Adam as an optimizer with a learning rate of 0.0001. To find out the right number of epochs, we applied early stopping.

For the random forest classifier, we set the number of trees to 100, minimum sample split to 2, minimum samples in leaf to 1, bootstrap to true, and criterion to Gini.

We set the parameters of the xgboost classifier as follows: eta to 0.3, min-child-weight to 1, max-depth to 6, and scale-pos-weight to 1.

We trained all three classifiers on the same learning dataset. For ensemble prediction, we used each classifier to predict an author, then calculated the average of the three output vectors and determined the most probable author. When there were two or more authors with the same probability, we chose randomly one of them as the predicted author.

4.2 Email Detective

We implemented the Email Detective algorithm without the email header stylistic features, considering no such data are available for datasets other than emails. We set the specification as described in [7] to an extent to which model parameters were specified. The authors of [7] calculated the word2vec vector representation with dimensions set to 256; the iter and window parameters set to 5. The BiLSTM layer was set to a full output sequence; a maxpool layer reduces

² https://github.com/karasekadam/authorship_identification

Table 2: Ensemble model experiment results for Techcrunch dataset

Techcrunch	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.9624	0.9275	0.7568
Random Forest	0.904	0.8517	0.6096
XGB classifier	0.9616	0.915	0.7536
MLP	0.9728	0.943	0.7504
Training time	894s	1229s	1142s

the output size. After a 0.5 dropout, a softmax layer follows. The classification part consists of a dense layer with 256 nodes, a dropout layer set to 0.5, and a softmax layer as output.

There were parameters not detailed in the original paper. We decided to use global max pooling 1D to reduce batch dimension for the max-pooling layer. The authors of [7] did not specify the length of the input text, so we experimented with different setups and decided to limit the input length to 10000 characters due to a need for a faster training time and limited memory.

4.3 BertAA

We used the TensorFlow hub to obtain the pre-trained BERT model and downloaded `bert_en_cased_L-12_H-768_A-12`. We transformed the input text to the pre-trained vector representation that was part of the downloaded model. A dense layer classified the output of the BERT model with a softmax activation function. All parameters were set to trainable, and we ran the experiment with early stopping.

5 Evaluation

We evaluated the model using *accuracy* and measured the training time. We performed a 72/18/10 split into training/validation/test.

We used a computer with Ubuntu 22.04 LTS, two Tesla T4 GPUs, 65GB RAM, and 32 32-core Intel Xeon Silver 4110 CPU for the experiment.

The count vector’s feature size in the Telegram dataset’s ensemble model was 4500 to 6000, with numbers in the upper of the interval with more authors. For the Techcrunch dataset, the count vector dimension was between 44000 and 55000. Furthermore, the count vector length for the Enron dataset was around 26000. The count vector size represents the training corpus’s unique words without English stopwords. As we expected, the size of the count vector was larger with more extensive datasets and longer texts.

6 Results

We only evaluated the count vector ensemble model described in Section 4.1. The experiment results for Techcrunch, Telegram, and Enron are shown in

Table 3: Ensemble model experiment results for Telegram dataset

Telegram	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.34	0.2062	0.0896
Random Forest	0.344	0.2062	0.0902
XGB classifier	0.308	0.1985	0.0885
MLP	0.31	0.2338	0.0766
Training time	52s	78s	190s

Table 4: Ensemble model experiment results for Enron dataset

Enron	$k = 5$	$k = 10$	$k = 25$
Ensemble	0.9675	0.9339	0.8417
Random Forest	0.9625	0.9228	0.8171
XGB classifier	0.9395	0.8961	0.8142
MLP	0.968	0.9234	0.8057
Training time	902s	1323s	1591s

Tables 2, 3, and 4, respectively. The tables show results for different numbers of authors.

The BertAA and Email Detective experiment was not finished by the time this paper was written. All measured results will be presented at the Raslan Conference 2023.

It can be seen the ensemble robustness shows itself when used on a larger number of authors. With enough features, the MLP model outperformed the ensemble model on the Techcrunch dataset with 5 and 10 authors and the Enron dataset with 5 authors. Furthermore, the random forest algorithm is probably better at classifying data with fewer features as it outperformed other models on the Telegram dataset, which has several times lower feature space than the other two datasets.

A possible explanation for the Telegram dataset’s drop in accuracy is the text’s nature. The texts are mutually very similar in topic and style, so it is difficult to distinguish the author by the words they have used.

7 Conclusion and Future Work

The aim of this paper was to examine recent approaches to authorship identification. We selected three of them and three datasets. We re-implemented each approach and evaluated it on all three datasets to see how reproducible the original paper is and how robust the approach is.

The results published in [2] were not fully reproducible since the authors did not publish details about MLP classifier architecture. In addition, the evaluation dataset was different from ours. Despite these conditions, the evaluation on the three datasets has shown that the approach is somewhat robust. Mainly, MLP

contributes the most to high accuracy in the case of a smaller number of potential authors. On the other hand, random forests are more accurate with a higher number of authors. Apparently, the text length does not matter much, but the number of features does.

In the near future, we plan to evaluate the re-implementations of the Email Detective and BertAA. In the case of Email Detective, we cannot expect the same results since the re-implementation does not take the email header into consideration. At least, we could estimate how much the email header classification contributes to the overall accuracy.


Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Abbasi, A., Chen, H.: Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace. *ACM Trans. Inf. Syst.* **26**(2) (apr 2008). <https://doi.org/10.1145/1344411.1344413>, <https://doi.org/10.1145/1344411.1344413>
2. Abbasi, A., Javed, A.R., Iqbal, F., Jalil, Z., Gadekallu, T.R., Kryvinska, N.: Authorship identification using ensemble learning. *Scientific Reports* **12**(1) (2022). <https://doi.org/10.1038/s41598-022-13690-4>
3. Anton: Crypto telegram groups (Feb 2021), <https://www.kaggle.com/datasets/aagghh/crypto-telegram-groups>
4. Balbo, T.: Techcrunch posts compilation (Oct 2016), <https://www.kaggle.com/datasets/thibalbo/techcrunch-posts-compilation>
5. Cohen, W.W.: Enron email dataset (2015), <https://www.cs.cmu.edu/~wcohen/>
6. Fabien, M., Villatoro-Tello, E., Motlicek, P., Parida, S.: BertAA : BERT fine-tuning for authorship attribution. In: Bhattacharyya, P., Sharma, D.M., Sangal, R. (eds.) *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. pp. 127–137. NLP Association of India (NLP AI), Indian Institute of Technology Patna, Patna, India (Dec 2020), <https://aclanthology.org/2020.icon-main.16>
7. Fang, Y., Yang, Y., Huang, C.: EmailDetective: An Email Authorship Identification And Verification Model. *The Computer Journal* **63**(11), 1775–1787 (07 2020). <https://doi.org/10.1093/comjnl/bxaa059>, <https://doi.org/10.1093/comjnl/bxaa059>
8. Nicholson, C.V.: A beginner's guide to word2vec and neural word embeddings, <https://wiki.pathmind.com/word2vec>
9. Nirkhi, S., Dharaskar, R.: Comparative Study of Authorship Identification Techniques for Cyber Forensics Analysis. *International Journal of Advanced Computer Science and Applications* **4** (12 2013). <https://doi.org/10.14569/IJACSA.2013.040505>
10. Nirkhi, S., Dharaskar, R., Thakare, V.: Authorship Verification of Online Messages for Forensic Investigation. *Procedia Computer Science* **78**, 640–645 (2016). <https://doi.org/https://doi.org/10.1016/j.procs.2016.02.111>, <https://www.sciencedirect.com/science/article/pii/S1877050916001137>, 1st International Conference on Information Security & Privacy 2015

11. Shetty, J., Adibi, J.: The Enron Email Dataset Database Schema and Brief Statistical Report. Tech. rep., University of Southern California (01 2004)

Augmenting Stylometric Features to Improve Detection of Propaganda and Manipulation

Radoslav Sabol and Aleš Horák 

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

Abstract. Identification of manipulative techniques in newspaper texts allows an informed reader to cope with the text content without being negatively influenced.

In this paper, we present new developments in using stylometry to support a deep learning neural network model in labelling newspaper articles for the presence of specific manipulative techniques. We also evaluate all stylometric features in 16 groups and improve the manipulation detection results in 15 of 17 techniques.

Keywords: propaganda detection, manipulative techniques, Propaganda dataset, stylometry

1 Introduction

Propaganda newspaper articles employ specific rhetorical figures to drive the readers opinion or to manipulate them, for example fabulation, labelling or demonization [10]. Detecting a presence of these devices in the text can be a strong indication that the text embodies malicious or ulterior motives.

In the previous work [12,11], a new deep learning approach that combines transformer-based large language model analysis with stylometric features has been introduced. The combination allowed to improve state-of-the-art results with the Propaganda benchmark dataset [1] for 14 of 17 manipulative techniques.

In the current paper, we present the new developments in the stylometric features set and evaluate feature group assets by a series of ablative sets. The final results reveal further enhancement of the results for 15 techniques.

1.1 Related Works

There has yet to be a genuine consensus within the scientific community on the optimal and universal set of stylometric features to be used in style analysis tasks. The choice usually depends on the currently solved task and applied

classification algorithm [9]. For the best variety, numerical features that reflect the author’s writing style are tailored from multiple levels of linguistic analysis.

Syntactic features attempt to exploit the sentence structure. Straightforward and common approaches make use of punctuation mark frequency, placement, and sentence lengths. More complex methods involve the extraction of information from the syntactic trees. Feng et al. [4] use two kinds of syntactic features for deception detection. Shallow syntactic features utilize the part of speech tags, while deep features encode the tree as a probabilistic context-free grammar. It was shown that the syntactic features do not outperform other feature types by themselves; they still carry viable information that can be utilized in conjunction with different feature types [7].

2 Stylometric Feature Set

The following section describes modifications to the previous stylometric feature set [12] to provide more detailed insight from various levels of linguistic analysis. Table 1 briefly explains the current state of implemented features, while other subsections will present the proposed changes in detail.

Table 1: Overview of the updated set of stylometric features. Features highlighted in bold are brand new additions to the old set. The feature highlighted in italic was significantly modified from the previous iteration.

Feature Type	# features	Language Independent
Word Length	137	✓
Sentence Length	177	✓
Word Repetition	140	
Word Class <i>n</i> -Grams	514	
<i>Morphological Tags n-Grams</i>	1,434	
Letter Casing	494	✓
Word Suffixes	425	✓
Word Richness	6	✓
Stopwords	600	✓
Punctuation	147	✓
Typography	111	✓
Character <i>n</i> -Gram Distribution	6,550	✓
Emoticons Presence	28	✓
Readability Metrics	4	
Structural Tree Characteristics	180	
Dependency relations <i>n</i>-Grams	3,208	
Total	14,155	

2.1 Syntactic Features

The following subsection covers new stylometric feature extractors that describe sentence structure from dependency trees. This information is subtly described using several existing features (sentence lengths, punctuation frequencies).

However, dependency trees allow for unique details that may improve the current feature set.

First, a dataset needs to be augmented by an additional support object. Before this work, the list of objects for each document was the following:

- **text**: the original plaintext document
- **lemmas**: a list of tokens and lemmatas
- **morphology**: morphological annotations from majka [14] and desamb [15]

To utilize the syntactic information, we create a new support object called **syntax**, which contains dependency trees for each document sentence. The trees are extracted using **UDPipe** [13], allowing a straightforward switch to other languages when necessary.

Structural Tree Characteristics The first group of features ignores all syntactic relations and observes only the structure of a tree. There are currently three feature extractors implemented:

1. **depth of the tree** (40 features)
 - for each tree, compute the longest path from the root node to any of the leaf nodes
2. **branching factors** (40 features)
 - for each non-leaf node of every tree, the number of children
3. **tree width** (100 features)
 - for each tree, compute the number of leaf nodes

The resulting vectors correspond to the relative frequency distributions of depths/widths/branching factors. To better convey the notion of adjacency between individual values, additional bins of size 2-3 are added to capture close values. The bins are illustrated in Figure 1.

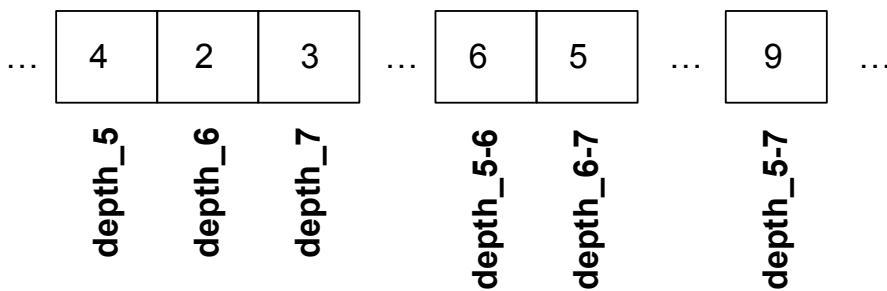


Fig. 1: The example of value binning for tree depth feature extractors. For illustrative purposes, the features are not normalized.

Relation-based Features The second set of features focuses on part-of-speech tags in the tree's nodes and the relations between them. We propose three types of characteristics, where each feature extractor reveals different information about the relationships in the dependency tree.

1. **Node n -grams** (735 features)
 - 2–4-grams, unigrams are essentially identical to word-class unigrams already present in the set
 - n -gram is constructed as an ascending path of node labels
2. **Relation n -grams** (1,415 features)
 - 1–4-grams using an ascending path of edge labels
3. **Complete n -grams** (1,058 features)
 - 2–4-grams, where n -gram is a path containing both node and edge labels (however, only node labels count towards an n -gram)

The training corpus calculates the list of allowed n -grams in advance. The preparation extracts all relevant n -grams from the documents and constructs the vector only using instances present in at least 1% of the documents. An example tree along with extracted n -gram is shown in Figure 2.

2.2 Readability Measures

The readability extractors present a group of four numerical features, where each element corresponds to a readability measure extracted from the input document. All of the metrics depend on the number of words, sentence lengths, and syllables, making them straightforward to adapt to other languages.

The **Flesch Reading Ease** [5] is considered to be one of the most commonly used and reliable readability metrics [8]. The values are scaled from 0 to 100, where higher values indicate that the text is easier to understand. The score is computed in the following way:

$$FLESCH = 206.835 - (1.015 * \bar{S}) - (84.6 * \bar{W})$$

Where \bar{S} is the average number of syllables per word (or the total number of syllables divided by the number of words); similarly, \bar{W} represents the average number of words per sentence. The final score is divided by 100 to make the domain consistent with other features.

The **Gunning Fog formula** is computed from a random sample of 100 sentences [2]. The resulting index approximates the years of formal education required to comprehend the text easily. The following formula computes it:

$$GUNNING_FOG = 0.4 * \bar{S} * H$$

Where H stands for a percentage of complex words. We consider a word to be complex when its lemma is more than two syllables long. The final index is normalized according to the most difficult article within the training corpus.

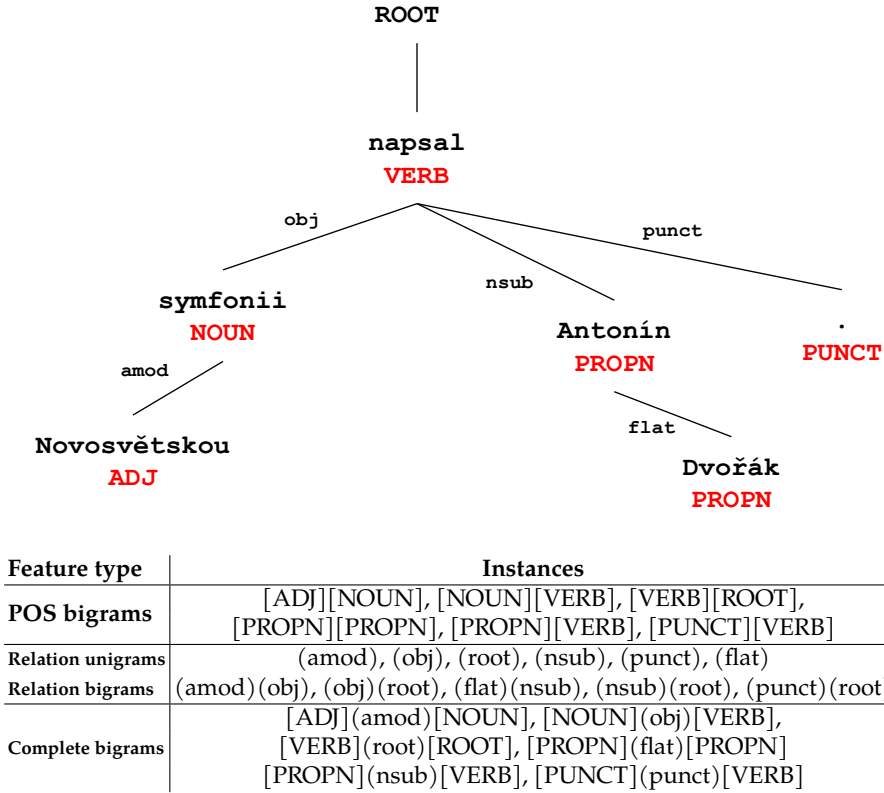


Fig. 2: An example dependency tree. The table below the tree lists the extracted n -grams from the sample tree.

The **McLaughlin's SMOG formula** is considered a more easily computed substitute for the Gunning Fog Index [6]. The interpretation of values remains the same, while the index is computed as:

$$SMOG = 3 + \sqrt{H}$$

Where H is the percentage of hard words in a random sample of 30 words.

Last but not least, the **FORCAST formula** uses an opposite approach where "easy" words are counted instead of the difficult ones [3]. It is computed as follows:

$$FORCAST = 20 - (E/10)$$

Where E is the number of single-syllable lemmas in a 150 word sample.

2.3 Limiting Morphological Tags

The original stylometric feature set included 10,000 features for various n -grams of morphological tags extracted from the training corpus. The feature amount was fixed to the most common morphological tags n -grams dependent only on n , and did not factor in the actual frequencies in the training corpus. This method led to the feature sets containing highly improbable n -grams where it is unclear whether the tags yield any significance or can be discarded as random noise. The noisiness can be observed from the previous works where the tags were frequently the least significant feature of the feature set via ablation tests [11].

The solution includes more strict limits for selecting morphological n -grams based on document frequencies. There is no strict limit on how many morphological n -grams need to be present in the feature set; however, it is required that the n -gram is present in at least 2% of the training documents. This method ensures that the feature vector will not contain improbable phenomena.

3 Experiments

The performed experiments focus on two aspects of the stylometric feature set: the importance of individual feature extractors for the current task and the overall performance of the modified stylometric feature set against the previous one. Both goals are benchmarked using the Propaganda dataset [1].

The Propaganda dataset includes 17 attributes, where 8 of them are manipulative techniques commonly used in misinformative news domains and thus are sensitive to style analysis. The remaining attributes focus on the properties or specific phenomena of the article, like genre, topic, or the writer's opinion of Russia. The dataset is split into train and test partitions, where the test partition contains a balanced sample of approximately 1000 documents as in [11].

The benchmarking uses gradient-boosted decision trees (GBDT) due to their reasonable performance and running times. Each experiment is repeated three times (indexed as i), with seeds fixed to $40 + i$ for results to be reproducible. The size of the ensemble is limited to 100 trees. The Weighted F1 is used as the performance metric to factor in the imbalance in the dataset.

3.1 Feature Selection

This experiment aims to measure the importance of individual feature extractors and select the most appropriate feature for each extractor. For this purpose, all feature extractors are grouped into 16 categories. First, the performance on the full feature set is measured as a base. Then, one of the categories is selected and removed from the complete feature set. Finally, a new model is trained on the reduced feature set, and a difference in weighted F1 from the complete feature set is measured. This process is applied to all feature categories. Similarly to the base experiments, the ablation tests are repeated three times on different seeds to estimate the difference in performance better.

For the fairness of comparison, a development set of approximately 20% of instances from the original training set is created. After the least significant features are determined, new results will be computed using the refined feature set with the feature of least significance removed.

4 Results and Discussion

In this section, the comparison of results on the benchmarking dataset Propaganda is performed. Detailed results of the ablation tests are discussed to better understand the interaction of stylometric features with the classes in the dataset.

Table 2: Summary of results for all attributes of the Propaganda dataset. The first and second row compares the old and new features sets. The third row describes the weighted F1 after the removal of the least significant feature group.

	argumentation	blaming	demonization	emotions	fabulation	fear-mongering	labelling	relativization
Old Features	68.54	71.67	95.60	80.69	79.72	90.02	82.13	92.48
New Features	69.04	71.72	95.75	80.56	80.21	90.73	82.78	92.56

	genre	location	sentiment	scope	topic	expert	opinion	Russia	source
Old Features	95.47	69.42	79.45	86.65	58.21	71.44	87.41	80.61	67.07
New Features	95.71	70.47	81.19	85.51	59.37	73.75	87.74	81.11	67.54

4.1 Comparison with the Previous Feature Set

A comparison of results between old and new feature sets can be seen in Table 2. The table also contains a third row with weighted F1 corresponding to the model trained on new features with the least important feature category removed.

Overall, the weighted F1 was improved by the refined features for almost every attribute of the Propaganda dataset. The only exceptions are emotions, where the new features perform worse by 0.13%, which is a difference that can be attributed to random error in measurement. A much more significant loss of 1.14% can be observed with the *scope* attribute. Even when removing the least significant feature (word class *n*-grams), the performance cannot be matched with the old feature set. In this instance, changing the morphological features could harm this attribute’s performance.

The greatest improvement was achieved for the *expert* attribute (2.33%) that could be explained by proper noun relations that were extracted as part of dependency tree analysis. A similar argument could be used to describe the enhancement for the *labelling* attribute, as propaganda labels usually consist of literal or metaphorical comparisons that can be extracted using relations in the dependency trees.

4.2 Feature Selection Results

The heatmap in Figure 3 shows the results for the performed ablation tests. Brighter colors indicate that the selected feature subset has higher importance, while the dark ones suggest that the feature is either not essential or even performance-degrading.

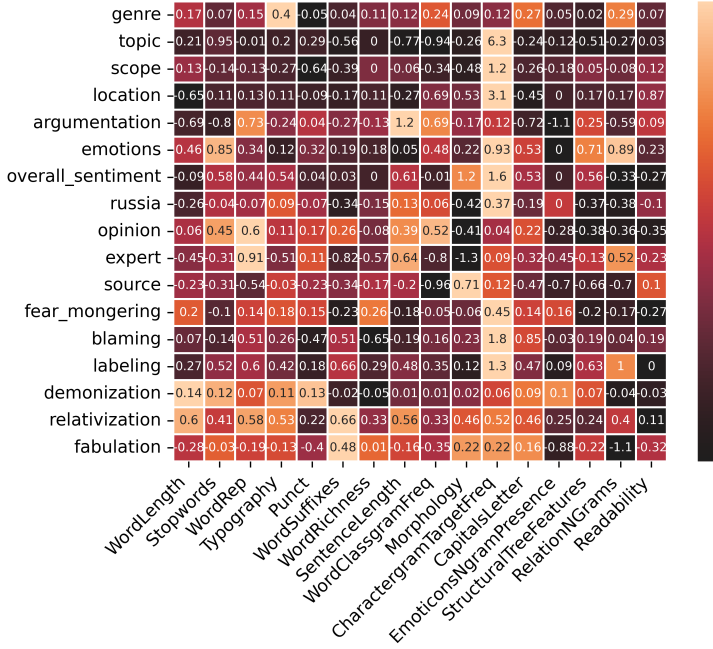


Fig. 3: Heatmap of feature group importances. The color map is normalized in a row-wise fashion to highlight notable importances.

The highest observed feature importances are 6.3% and 3.1% for Character n -Grams in *location* and *topic* attributes. These two attributes are heavily tied with the semantics of the text, and sufficiently large character n -grams (in this case, 5-grams) can capture keywords that are tied with these attributes. As *location* and *topic* should not be affected by the writing style, having semantic cues captured within character n -grams is vital for improving accuracy for such attributes.

Another notable importance is 1.2% for sentence lengths in *argumentation* detection. A possible explanation is that when the author uses more complex reasoning, his sentences are usually longer as they contain sub-sentences which logically follow-up the argument.

The new features presented in this paper positively contribute to *labelling* detection (where relation *n*-grams are the second most important features), *expert*, and *emotions*.

To finish off the experiment, we remove the least important feature for each attribute, and evaluate a new model. The results can be observed on Table 3.

Table 3: Re-evaluation of GBDT with the least significant feature removed. **Importance** column refers to the difference in weighted F1 as shown on the heatmap. **F1** refers to the Weighted F1 performance metric of the new model with **bold** values referring to the best numbers for each attribute. **Diff** is a difference against the model trained on a complete feature set.

Attribute	Removed Feature	Importance	F1 (%)	Diff
Argumentation	Emojis	-1.1	69.34	0.30
Blaming	Word Richness	-0.65	71.33	-0.39
Demonization	Word Richness	-0.05	95.75	0.00
Emotions	Emojis	0	80.40	-0.16
Fabulation	Relation <i>n</i> -grams	-1.1	80.52	0.31
Fear-mongering	Readability	-0.27	90.63	-0.10
Labeling	Readability	0	82.09	-0.69
Relativization	Readability	0.11	92.64	0.08
Genre	Punctuation	-0.05	95.52	-0.19
Location	Word Lengths	-0.65	69.75	-0.72
Sentiment	Relation <i>n</i> -grams	-0.33	80.60	-0.59
Scope	Punctuation	-0.64	86.04	0.53
Topic	Word Classes	-0.94	60.05	0.68
Expert	Morphology	-1.3	73.83	0.08
Opinion	Morphology	-0.41	87.95	0.21
Russia	Morphology	-0.42	80.87	-0.24
Source	Word Classes	-0.96	67.07	-0.47

If the importance is low enough, it is possible to further improve the performance by removing the feature extractor. The improvement holds mainly for the *Topic* and *Scope* attributes. However, this is not guaranteed, as the features removed for the *Source* attribute also have a low importance, but the performance degrades significantly.

There is no significant correlation between the feature importance and the difference in performance. There may be more variables involved, like the actual number of removed numerical features and the choice of the learning algorithm.

5 Conclusion and Future Work

We have presented an extension of stylometry features used, in conjunction with large language models, for identification of 17 different manipulative techniques and propaganda reflection techniques as employed in newspaper texts. By adding syntactic features, readability metrics and by adjusting the previous morphological features, the manipulation detection models are improved with 15 of the 17 text attributes.

In the future work, we plan to accomplish further steps in adjusting the detailed best sets of features for each attribute and to tune the feature weights by comparing their values in propagandistic and standard newspaper texts.



Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2023062. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Baisa, V., Herman, O., Horak, A.: Benchmark dataset for propaganda detection in Czech newspaper texts. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 77–83. INCOMA Ltd., Varna, Bulgaria (Sep 2019). https://doi.org/10.26615/978-954-452-056-4_010, <https://aclanthology.org/R19-1010>
2. Bogert, J.: In defense of the fog index. *The Bulletin of the Association for Business Communication* **48**(2), 9–12 (1985). <https://doi.org/10.1177/108056998504800203>, <https://doi.org/10.1177/108056998504800203>
3. Caylor, J.S., Others: Methodologies for determining reading requirements of military occupational specialties (1973)
4. Feng, S., Banerjee, R., Choi, Y.: Syntactic stylometry for deception detection. In: Li, H., Lin, C.Y., Osborne, M., Lee, G.G., Park, J.C. (eds.) Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 171–175. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), <https://aclanthology.org/P12-2034>
5. Flesch, R.: A new readability yardstick. *Journal of applied psychology* **32**(3), 221 (1948)
6. Harry, G., Laughlin, M.: SMOG Grading – a new readability formula. *The Journal of Reading* (1969), <https://api.semanticscholar.org/CorpusID:9571753>
7. Hollingsworth, C.: Syntactic stylometry: Using sentence structure for authorship attribution (2012), <https://api.semanticscholar.org/CorpusID:15376304>
8. Klare, G.R.: The measurement of readability: useful information for communicators. *ACM Journal of Computer Documentation (JCD)* **24**(3), 107–121 (2000)
9. Lagutina, K., Lagutina, N., Boychuk, E., Vorontsova, I., Shliakhtina, E., Belyaeva, O., Paramonov, I., Demidov, P.: A survey on stylometric text features. In: 2019 25th Conference of Open Innovations Association (FRUCT). pp. 184–195 (2019). <https://doi.org/10.23919/FRUCT48121.2019.8981504>

10. Miles, C.: Rhetorical Methods and Metaphor in Viral Propaganda. In: Bains, P., O'Shaughnessy, N., Snow, N. (eds.) *The SAGE Handbook of Propaganda*, pp. 155–170. SAGE Publications (2019)
11. Sabol, R.: Propaganda Detection using Stylometric Text Analysis. Master thesis, Masaryk University, Faculty of Informatics, Brno (2023), <https://is.muni.cz/th/b83ai/>
12. Sabol, R., Horák, A.: Manipulative Style Recognition of Czech News Texts using Stylometric Text Analysis. In: *Recent Advances in Slavonic Natural Language Processing, RASLAN 2022*. pp. 191–199 (2022)
13. Straka, M., Straková, J.: Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. pp. 88–99. Association for Computational Linguistics, Vancouver, Canada (August 2017), <http://www.aclweb.org/anthology/K/K17/K17-3009.pdf>
14. Šmerk, P.: Fast Morphological Analysis of Czech. In: Sojka, P., Horák, A. (eds.) *Third Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2009*. pp. 13–16. Masaryk University (2009)
15. Šmerk, P.: K počítačové morfologické analýze češtiny (in Czech, Towards Computational Morphological Analysis of Czech). Ph.D. thesis, Faculty of Informatics, Masaryk University (2010)

Predicting Style-Dependent Collocations in Russian Text Corpora

Lada Petrushenko  and Olga Mitrofanova 

Saint Petersburg University
Universitetskaya emb. 7-9
199034 Saint Petersburg, Russia
ladapetrushenko@gmail.com, o.mitrofanova@spbu.ru

Abstract. The paper presents the results of experiments on the development of distributional language models for predicting acceptable collocations of the ADJ+NOUN type. The models are trained on stylistically diverse Russian corpora (news, popular science, fiction, poetry). The evaluation of the models allows us to determine optimal parameters for collocation prediction and explore linguistic features of predicted collocations.

Keywords: predictive distributional models, collocations, Russian corpora.

1 Introduction

Since the emergence of Word2Vec in 2013, predictive distributional language models have become the preferred tool for dealing with semantic NLP tasks. With their help, researchers can now process huge amounts of text data at a faster rate and overcome technical limitations while working with large text collections. Moreover, predictive models have proven to be more effective at representing semantic relations between tokens compared to count-based models. Over the past ten years, a number of resources with pretrained predictive models have emerged, making it possible for any user to access the required data and investigate language phenomena both observable and unseen in corpora.

Such language models can be used to predict both paradigmatic and syntagmatic relations between tokens. To date, most Russian researchers focused on the investigation and description of paradigmatic relations: e.g., semantic similarity evaluation and taxonomy enrichment tasks were under discussion in RUSSE contests, organized within the conference on Computational Linguistics and Intellectual Technologies “Dialogue” [1]. Nevertheless, syntagmatic relations underlying lexical constructions of various types are described less thoroughly within the framework of distributional semantic models for Russian, except for a few projects, the CoCoCo database [2] and the DSM-Calculator [3] are among them. Predicting collocations of a predefined type is useful for a number of NLP tasks such as text summarization, text generation, sentiment analysis, etc., as it can improve the quality of task implementation.

The study is based on the assumption that linguistic properties of predicted collocations are determined by features of language models. Thus, the core experiments aim to define the optimal parameters that provide training of non-contextualized distributional models for predicting acceptable collocations: vector space dimensionality, context window size, corpus preprocessing, dictionary filtering, similarity measures, etc. The study focuses on predicting a particular type of collocations for Russian texts of different styles. The description of predicted style-dependent collocations, developed during our study, fills the gaps in Russian NLP. It allows us to obtain novel results that are relevant for text classification based on construction identification [4] and style transfer [5].

The paper is structured as follows: the section "Related work" contains the theoretical foundations of our study, the section "Experimental design" describes the linguistic data used in experiments and explains the experimental procedures, the section "Results" provides an overview of the optimal parameters for collocation prediction and the linguistic features of predicted collocations, and the section "Conclusion and further research" summarizes the achieved results and outlines future work.

2 Related Work

The idea of a lexical construction as a core unit of language can be traced back to the Construction Grammar (CxG) theory developed by Ch. Fillmore [6]. According to Ch. Fillmore, a lexical construction is a sequence of lexical units in which some components define the surrounding context while others serve as supplementary elements [7]. CxG emphasizes certain characteristics of lexical constructions, such as their semantic, syntactic, and pragmatic nature, as well as the potential for idiomatic meanings. Lexical constructions are usually classified as regards their idiomaticity and compositionality [8,9,10,11]. Following [11], collocations are treated as lexical constructions with partially restricted use of its components.

There are three main approaches to collocation extraction: count-based (statistical) approach involves association measures (e.g., PMI, t-score, Log Likelihood, Chi-square, etc.) and vectorization models (e.g., TF-IDF, LSA, HAL, COALS, etc.); hybrid approach, which relies on both linguistic and statistical information, is implemented in techniques for extracting lexical-grammatical patterns (e.g., keyphrase extraction algorithms like RAKE, KEA, Topia, etc.); predictive approach is implemented in distributional semantic models of dense word embeddings. Predictive models are represented by non-contextualized or static Word2Vec-type models and contextualized Transformer-based models. Among non-contextualized models, such as Word2Vec [12,13] and FastText [14,15] can be highlighted. As for contextualized models, it is worth mentioning BERT [16], ELMo [17] and contemporary developments. In this research, we focus on the first type of models due to the fact that it is more challenging to control prediction results when working with BERT-like models [18].

Predictive models have proved to be more effective at detecting word similarity compared to count-based models in a number of tasks such as synonym detection, measuring semantic relatedness, concept categorization, etc. [19]. Research has also shown that these models can be applied to predicting specific types of collocations, such as constructions consisting of a verb and a noun [20], constructions with an attributive meaning [21], collocations expressing lexical functions [22]. In our study, we focus on the task of predicting style-dependent collocations of ADJ+NOUN type.

3 Experimental Design

In our research, we used segments of Taiga [23] and Lib.ru.sec [24] corpora consisting of stylistically diverse Russian texts: news, popular science, fiction, poetry. We conducted two sets of experiments: the first aimed at detecting the optimal parameters for collocation prediction, and the second aimed at describing the linguistic features of the predicted collocations.

For the first experiment, the subcorpora of the following size were used:

- **Fontanka** (the news subcorpus of Taiga) comprises 73,140,388 tokens and 3,885,119 sentences (the entire subcorpus was taken for both the experiments);
- **Nplus1** (the Taiga subcorpus of non-fictional (popular-science) texts) comprises 1,667,938 tokens and 72,002 sentences (the entire subcorpus was taken for both the experiments);
- **Stihi_ru** (the Taiga subcorpus of poems) comprises 5,986,693 tokens and 421,956 sentences (the first 50,000 texts were taken for the first experiment);
- **Lib.ru.sec** (the Taiga subcorpus of fiction texts) comprises 9,669,140 tokens and 677,134 sentences (the first 100 texts were taken for the first experiment).

Due to technical limitations, it was not possible to process the complete versions of *Stihi_ru* and *Lib.ru.sec*, but the representative subcorpora of both of them were taken.

For working with the data of *Fontanka* and *Nplus1* in the first experiment, we relied on the predefined annotation provided by the authors of the dataset. This implies morphosyntactic annotation performed in terms of Universal Dependencies (UD) [25]. *Stihi_ru* and *Lib.ru.sec* were downloaded in *.txt format and then annotated by us with *spacy_udpipe* [26]. After preprocessing, all the data was presented in the CoNLL-U format [27].

For the second experiment, we annotated all the subcorpora with the *pymorphy2* morphological tagger [28] to test whether this type of annotation could increase the quality of predictions. After analyzing the predictions of the first set of the “best” models, we added the following restrictions to the algorithm of handling the data:

- token length should be more than 2 characters;

- tokenized sentence length should be more than 2 tokens;
- annotation in terms of the pymorphy2 tagset should be transformed into UD annotation.

As a result, for the second experiment we worked with the following dataset:

- **Fontanka** (the news subcorpus of Taiga) comprises 41,234,011 tokens and 3,611,338 sentences (the entire subcorpus was taken for both the experiments);
- **Nplus1** (the Taiga subcorpus of non-fictional (popular-science) texts) comprises 1,328,657 tokens and 90,313 sentences (the entire subcorpus was taken for both the experiments);
- **Stihi_ru** (the Taiga subcorpus of poems) comprises 5,961,406 tokens and 703,358 sentences (the first 100,000 texts were taken for the second experiment);
- **Lib.ru.sec** (the Taiga subcorpus of finction texts) comprises 31,591,065 tokens and 3,791,616 sentences (the first 1000 texts were taken for the second experiment).

In both experiments, we trained a set of Word2Vec and FastText models and validated the results with the help of pseudo-disambiguation procedure, a common approach to testing the quality of predictions [29,30]. Under this approach, the test data comprises combinations of three tokens:

- target word: e.g., *день* (*day*);
- candidate word that can form a lexical construction together with the target word due to their co-occurrence in the corpus: e.g., *летний день* (*summer day*);
- candidate word that can form a lexical construction but does not occur with the target unit in the specific corpus or a candidate word that cannot form a lexical construction with the target token at all: e.g., **железный день* (**iron day*).

Correct collocations for pseudo-disambiguation were chosen on the basis of whether they occurred in all 4 subcorpora at least 8 times. The incorrect collocations were chosen according to relative frequencies: if the relative frequency of occurrence for an incorrect collocate was lower than the relative frequency of occurrence for a correct collocate in all the subcorpora, such incorrect collocate was taken into account in our data. As a result, we obtained a set of 155 combinations of target word-correct collocate-incorrect collocate, which we used to evaluate the performance of the models. The evaluation was conducted on the same data regardless of the genre of the text on which the model was trained.

The models were trained in gensim [31] with all possible combinations of the following 6 parameters:

- a metric to measure the degree of similarity of vectors: Euclidean distance, squared Euclidean distance, cosine similarity, correlation of vectors;
- vector size: 100, 150, 200, 250, 300;

- context window size: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10;
- a metric that determines the threshold for word frequency, indicating which words should be considered by a model (`min_count`): 5, 10, 15;
- a metric that indicates the approach for computing a vector of contextual features when CBOW is chosen (`cbow_mean`): 0 (sum), 1 (mean);
- a metric that indicates the approach for sorting the vocabulary: 0 (none), 1 (by descending frequency).
- a metric responsible for limiting the usage of RAM when building the vocabulary (`max_vocab_size`): None, 30000, 60000; if the number of unique tokens is larger than the defined threshold, the model neglects low-frequency tokens.

The results of collocation prediction were evaluated with such metrics as precision, recall, and F1-score. The components for these metrics were calculated as follows:

- TP is the number of times when the model predicted a collocation correctly, and the tokens within the collocation semantically combine with each other;
- TN is the number of times when the model predicted a collocation incorrectly, and the tokens within the collocation do not semantically combine with each other;
- FN is the number of times when the model did not predict a collocation and could not differentiate between incorrect and correct collocates: the similarity measure between both pairs is similar up to two decimal places.

4 Results

4.1 Optimal parameters for collocation prediction

For investigation, we took the best 500 results for each model and analyzed the training parameters. The best results consist of combinations of model training metrics and were selected by sorting all the outcomes of a specific model based on the precision score metric in solving the pseudo-disambiguation task. Based on the first experiment and comparison of precision scores for all the results (cf. major best scores for Word2Vec models in Table 1), we made the following conclusions.

The parameters that have the most influence on prediction results are similarity measure, vector size, minimum word frequency for consideration by the model, and dictionary sorting parameter.

These parameters are organized in some sort of clusters where certain combinations of them have unchanging stable values while others can fluctuate. For example, the context window size and the approach to vector calculation did not affect the overall results within these clusters. The `cbow_mean` parameter could be either 0 or 1, without influencing the predictions. This observation applies to both Word2Vec models and FastText models.

It can be concluded that the correlation coefficient or cosine similarity measure are the most effective in determining the semantic relationships between

Table 1: Major training parameters for the best Word2Vec models.

Corpus	Precision %	metric	vector size	min_count	sorted_vocab	n_comb
Fontanka	62.58	cosine	250	15	0	20
NPlus1	89.03	cosine	250	15	1	20
		correlation	150	10	0	20
Stihi_ru	87.1	correlation	100	15	1	20
Lib.ru.sec	70.32	correlation	150	15	0	20
		cosine	250	15	1	20

collocation elements. The best 500 results of each model did not include those that used the Euclidean distance or squared Euclidean distance as the similarity measure. This finding supports the hypothesis of other researchers that these metrics are not as effective in capturing semantic properties compared to cosine similarity measure [32].

FastText models are better at predicting syntagmatic relations compared to paradigmatic relations. To prove this point, we took two nouns: *год* (*year*) and *исследование* (*study*), and examined how frequently adjectives appeared in the best 1000 predicted words for them. As a result, it turned out that there were no adjectives at all. For comparison, Word2Vec predicted 192 and 199 adjectives for these words respectively. Because of this, we decided not to use FastText models in our second experiment.

4.2 Linguistic features of predicted collocations

For the second experiment, we trained 4 Word2Vec models with the following best combinations of parameters:

- **Fontanka**: metric='cosine', size=200, min_count=15, sorted_vocab=0, window=any (default: 5), cbow_mean=any (default: 1);
- **Nplus1**: metric='cosine', size=150, min_count=15, sorted_vocab=0, window=any (default: 5), cbow_mean=any (default: 1);
- **Stihi_ru**: metric='cosine', size=150, min_count=10, sorted_vocab=1; window=any (default: 5), cbow_mean=any (default: 1);
- **Lib.ru.sec**: metric='cosine', size=100, min_count=15, sorted_vocab=0; window=any (default: 5), cbow_mean=any (default: 1).

We experimented with several words that were chosen from all the corpora randomly: *сайт* (*website*), *человек* (*man or human*), *научно-исследовательский* (*research or scientific-research*), *красивый* (*beautiful*), *день* (*day*), *система* (*system*).

We evaluated the results based on consistency coefficient A. The first evaluation procedure consists in the evaluation of consistency across our research models. For each of the mentioned words, we obtained 10 collocates from each model. Thus, having a total of 40 collocates for each word from the 4 models, except for *научно-исследовательский* (*scientific-research*) - for this word, we obtained 30 collocates as it was absent in the dictionary of the model trained on poetry. We performed pairwise comparisons of the results from Nplus1, Fontanka,

Lib.ru.sec, and Stihi_ru. The coefficient A was calculated as the number of overlapping predictions relative to all predictions (cf. Table 2). The predictions are considered overlapping if they appear in the predictions of at least two models.

Table 2: Evaluation of consistency among the models.

Target word	Repeating collocates	Consistency of predictions A
<i>сайт</i> (<i>website</i>)	<i>новостной, электронный, подробный</i> (<i>news, electronic, detailed</i>)	0,075 (3 repetitions per 40 collocates)
<i>человек</i> (<i>man or human</i>)	<i>верующий, больной, чужой, нищий</i> (<i>religious, sick, alien, poor</i>)	0,1 (4 repetitions per 40 collocates)
<i>красивая</i> (<i>beautiful</i>)	<i>блондинка, прелесть</i> (<i>blonde, charm</i>)	0,05 (2 repetitions per 40 collocates)
<i>система</i> (<i>system</i>)	<i>дистанционная, автоматическая</i> (<i>remote, automatic</i>)	0,05 (2 repetitions per 40 collocates)
<i>день</i> (<i>day</i>)	<i>выходной, летний, июньский, десятый, сегодняшний, бессонный</i> (<i>weekend, summer, June, tenth, today, sleepless</i>)	0,15 (6 repetitions per 40 collocates)
<i>научно-исследовательский</i> (<i>research or scientific-research</i>)	<i>машиностроение</i> (<i>mechanical engineering</i>)	0,033 (1 repetition per 30 collocates)

In some cases, the similarity value of a collocation predicted by one model could be twice as large compared to that of another one: cf. (*электронный сайт* (*website*), cosine = 0.31 vs. 0.61. This could be due to the fact that models show differences between the strength of connections within matching collocations. At the same time, the low number of overlapping predictions can be explained by topical differences of the corpora.

In the second experiment, we compared the results of predictions with the results from the Word Portrait project of The Russian National Corpus (RNC) [33]. Additionally, we made the same requests to two models from DSM-Calculator [3,34]: the model trained on the Russian Wikipedia dump in 2017 (referred to as DSM-Wiki), and the model trained on the Lib.ru corpus in 2017 with a context window size of 5 (referred to as DSM-Lib). Coefficient A is calculated based on the total number of predictions from the Nplus1, Fontanka, and Lib.ru.sec models for 6 target words, which amounts to 60 collocates, and the Stihi_ru model, which predicted 50 collocates.

- **Fontanka**: matches RNC in 3 predictions ($A = 0.05$); DSM-Wiki — in 7 ($A = 0.116$); DSM-lib — in 7 ($A = 0.116$);
- **Nplus1**: matches RNC in 3 predictions ($A = 0.05$); DSM-Wiki — in 5 ($A = 0.083$); DSM-lib — in 0 ($A = 0$);

- **Lib.ru.sec**: matches RNC in 2 predictions ($A = 0.033$); DSM-Wiki — in 2 ($A = 0.033$); DSM-lib — in 1 ($A = 0.016$);
- **Stihi_ru**: matches RNC in 5 predictions ($A = 0.1$); DSM-Wiki — in 1 ($A = 0.02$); DSM-lib — in 0 ($A = 0$).

The low number of matches once again shows that the model predictions strongly depend on the corpus style and main topics. At the same time, it can be unexpected that there is a low number of matches between the predictions of the Lib.ru.sec and DSM-lib models, since both had been trained on fiction texts. The differences between these models may be attributed to the fact that they were trained on different-sized datasets (around 9 million tokens and around 146 million tokens).

The predictions contain both established combinations, e.g. *официальный сайт* (*official website*), *выходной день* (*day-off*), *социальная система* (*social system*), etc.) and combinations that have represent terminological expressions, e.g. *бортовая система* (*on-board system*), etc.

The predicted constructions are mostly compositional and not idiomatic. The scientific-popular and news models perform worse in predicting constructions for the adjective *красивый* (*beautiful*) compared to fiction and poetic models. This can be explained by the fact that this adjective has a subjective interpretation that is less common in certain types of texts compared to fiction texts. There are instances of constructions where the meanings of the elements are not coordinated, for example, **красивое образование* (**beautiful education*), **красивая география* (**beautiful geography*), and **безлунный день* (**moonless day*). Such collocations are considered as anomalous.

5 Conclusion and future research

In this paper, we conducted several experiments on the prediction of noun phrases in Russian texts representing different writing styles: news, popular science, fiction, poetry. We analyzed a set of parameters and identified patterns that enable us to highlight specific parameters and approaches for predicting acceptable collocations unseen in the corpora. Multiple Word2Vec and FastText models were trained and evaluated, results leading to the conclusion that Word2Vec performs better in predicting syntagmatic relations, while FastText is better at predicting paradigmatic relations. Additionally, it is worth noting that such parameters as the association measure metric, vector size, minimum word frequency for model consideration, and dictionary sorting parameter play important roles in training the model for the prediction of noun phrases. Lastly, our experiments allowed us to observe the stylistic variation of collocations depending on the corpus type they were trained on.

The best models trained on stylistically diverse corpora are incorporated in the web-application “Construction Calculator” [35] developed on a Hugging Face platform. We plan to use the application for collocation generation in tests for studying Russian as a foreign language and for training collocation-

aware style-sensitive language models which are necessary in automatic style detection.

Acknowledgments. Research is performed with support of RSF grant № 21-78-10148 «Modeling the meaning of a word in individual linguistic consciousness based on distributive semantics».

References

1. RUSSE: Workshop on Russian Semantic Evaluation. <https://russe.nlpub.org/>, last accessed 31 Oct 2023
2. CoCoCo. <https://cococo.cosyco.ru/>, last accessed 31 Oct 2023
3. DSM-Calculator. <https://dsm-calculator.ru/>, last accessed 31 Oct 2023
4. Dubovik, A. Automatic text style identification in terms of statistical parameters. In: *Computer Linguistics and Computing Ontologies*. Issue 1, pp. 29-45 (2017)
5. RUSSE-2022: Detoxification. <https://www.dialog-21.ru/evaluation/2022/russe/>, last accessed 31 Oct 2023
6. Fillmore, C.J. Syntactic Intrusion and the Notion of Grammatical Construction. In: *Proceedings of the Eleventh Annual Meeting of the Berkeley Linguistics Society*, pp. 73–86 (1985)
7. Fillmore, C.J., Kay, P., O'Connor, M.C. Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone. In: *Linguistic Society of America*, Vol. 64, No. 3 (1988)
8. Vinogradov, V.V. *Selected Works: Lexicology and Lexicography* (1977)
9. Gak, V.G. On the Problem of Semantic Syntagmatics. In: *Language Transformations*, pp. 272–297 (1998)
10. Apresyan, Ju.D. *Selected Works*. Vol. 1. *Lexical Semantic: Synonymic Means of Language*. (1995)
11. Iordanskaya, L.N. Melchuk, I.A. *Sense and Compatibility in a Dictionary* (2007)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient Estimation of Word Representations in Vector Space. (2013a) <https://doi.org/10.48550/arXiv.1301.3781>
13. Mikolov, T., Yih, W., Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In: *Proceedings of NAACL-HLT 2013*, pp. 746–751. <https://aclanthology.org/N13-1090/>, last accessed 31 Oct 2023 (2013b)
14. Joulin, A., Bojanowski, P., Mikolov, T., Jegou, H., Grave, E. Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2979–2984 (2018) <https://doi.org/10.18653/v1/D18-1330>
15. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. Enriching Word Vectors with Subword Information. In: *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146 (2017) https://doi.org/10.1162/tacl_a_00051
16. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* – Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186 (2018) <https://doi.org/10.18653/v1/N19-1423>

17. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, Ch., Lee, K., Zettlemoyer, L. Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association of the Computational Linguistics: Human Language Technologies, Vol. 1, pp. 2227–2237 (2018) <https://doi.org/10.18653/v1/N18-1202>
18. Belyi, A.V. , Mitrofanova, O.A., Dubinina, N.A. Distributive Semantic Models in Language Learning: Automatic Generation of Lexical-Grammatical Tests for Russian as a Foreign Language. In: Proceedings of 2023 Corpus Linguistics Conference, 2023 (2023)
19. Baroni, M., Dinu, G., Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 238–247 (2014) <https://doi.org/10.3115/v1/P14-1023>
20. Kolesnikova, O., Gelbukh, A. A Study of Lexical Function Detection with Word2Vec and Supervised Machine Learning. In: Journal of Intelligent and Fuzzy Systems, pp. 1–8 (2020) <https://doi.org/10.3233/JIFS-179866>
21. Hartung, M., Kaupmann, F., Jebbara, S., Cimiano, Ph. Learning Compositionality Functions on Word Embeddings for Modelling Attribute Meaning in Adjective-Noun Phrases. In: 15th Meeting of the European Chapter of the Association for Computational Linguistics (EACL). <https://aclanthology.org/E17-1006/>, last accessed 31 Oct 2023 (2017).
22. Enikeeva E.V., Mitrofanova O.A. Russian Collocation Extraction based on Word Embeddings. In: Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference «Dialogue», pp. 52–64 (2017)
23. Taiga. https://tatianashavrina.github.io/taiga_site/, last accessed 31 Oct 2023
24. Lib.ru. <http://lib.ru/>, last accessed 31 Oct 2023
25. Universal Dependencies. <https://universaldependencies.org/>, last accessed 31 Oct 2023
26. Spacy_UDPipe. <https://pypi.org/project/spacy-udpipe/>, last accessed 31 Oct 2023
27. CoNNL-U. <https://pypi.org/project/conllu/>, last accessed 31 Oct 2023
28. pymorphy2. <https://pymorphy2.readthedocs.io/en/stable/>, last accessed 31 Oct 2023
29. Gale, W.A., Church, K.W., Yarowsky, D. Work on Statistical Methods for Word Sense Disambiguation. In: AAAI Fall Symposium on Probabilistic Approaches to Natural Language: Proceedings of the 29th Annual Meeting on Association for Computational Linguistics. <https://studylib.net/doc/13790396/work-on--statistical-methods-for--word-sense--disambiguation>, last accessed 31 Oct 2023 (1992)
30. Dagan, I., Marcus, S., Markovitch, S. Contextual Word Similarity and Estimation from Sparse Data. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 164–171 (1993) <https://doi.org/10.3115/981574.981596>
31. gensim. <https://radimrehurek.com/gensim/>, last accessed 31 Oct 2023
32. Rohde, D.L.T., Gonnerman, L.M., Plaut, D.C. An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence (2005)
33. RNC. <https://ruscorpora.ru/>, last accessed 31 Oct 2023
34. Bukia, G., Protopopova, E., Panicheva, P., Mitrofanova, O. Estimating Syntagmatic Association Strength Using Distributional Word Representations. Computational Linguistics and Intellectual Technologies: Proceedings of the Annual International Conference «Dialogue», pp. 112–122 (2016)

35. Construction Calculator. https://huggingface.co/spaces/ladapetrushenko/construction_calculator, last accessed 31 Oct 2023

Part III

Text Corpora

Semi-automatic Dictionary Creation for Czech

Using Automatisation to Create a Rapid Czech Dictionary

František Kovařík

Faculty of Arts, Masaryk University
Arna Nováka 1, 602 00 Brno, Czech Republic
frkov@mail.muni.cz

Abstract. There are many ways to make lexicographer's work faster and more accurate by using automatised and semi-automatic tools. In our project, we create a Czech dictionary using corpora and automatic tools, as well as subsequent manual annotations. We examine the semi-automatic method used in previous projects on different languages – its efficiency, accuracy and speed. This paper is to introduce the project, its preparations, its initial phases, as well as the goals of its research.

Keywords: Dictionary, manual, automatic, semi-automatic, headwords, annotation, revision, tag, lemma, POS, Czech

1 Introduction

Corpus-based and computational tools help lexicographers create dictionaries rapidly and accurately in more areas of expertise than ever before. This leads to easier dictionary creation and also helps the lexicographer skip the parts of creation which can be automatised, so they can focus on more difficult or interesting tasks. It also allows native speakers who are not professional linguists to help maintain accuracy and objectivity of dictionary entries while also boosting creation speed.

As described in previous papers [1,2], lexicographers of Lexical Computing have created dictionaries using a unique semi-automatic methodology. The methodology consists of multiple tools. Some are fully automatic. Some require manual inspection. Manual annotations were done by native speakers (we will refer to them as *annotators*). These were (with one exception) not professional linguists. The lexicographers themselves (we will refer to them as *coordinators*) didn't speak the languages of the created dictionary, and only had a limited knowledge of the language they were examining. In the course of multiple projects, the methodology continued to evolve and four dictionaries have been created: Lao, Tagalog, Urdu and Ukrainian. These consist of translations to English and/or Korean, as of sense distinction using thesaurus and/or pictures and the morphosyntactical behaviour.

The aim of this paper is to introduce a new rapid dictionary project called *Czech Dictionary Express*. We use the existing methodology to create a dictionary

of the Czech language and explore the possibilities, just as the downsides of this semi-automatic approach. We examine the first phases of the semi-automatic dictionary creation. We describe the main questions and problems that can arise within these phases.

2 Project phases and overlapping

We split the project into multiple phases.

The phases follow each other according to their number, but they can also overlap. For example, we can generate more headwords to the lexicon (a tool from Phase 0 – see section 3) while the annotators already annotate the headwords generated earlier (Phase 1 – see section 4) and even earlier generated and annotated headwords are being revised (Phase 2 – see section 5.1) and so on.

3 Preparation phase

In the preparation phase (which we call Phase 0), two objectives have been met:

3.1 Objective 1: Generating headword batches

First a list had to be generated of the lexicon that was going to be used in the dictionary. The list consists of headwords, i.e. lemma-POS couples (for example *místnost-noun*). These were taken from a corpus combining three of the largest Czech corpora, the csTenTen web corpora: csTenTen12, csTenTen19 and large part of csTenTen17. [3] The lexicon of the corpus is thus derived from relatively present-day Czech used on the web.

The lexicon was split into separate word batches, most containing 1 000 words.

Firstly, we only produced 30 batches, containing in total 15 000 most frequent unique words from the corpus. The batches 1–15 were identical to the batches 16–30 so inter-annotator agreement could be generated easily. In the process, we discovered the annotation went faster than expected, so we enlarged the lexicon to a final 80 000 words, producing over a hundred more batches.

One of the batches has been seven times multiplied and given to all the annotators so we could compare their annotations all together. All the other batches were only duplicated (e.g. batch 2 is identical to batch 17) and given to two different annotators. This was done to investigate the inter-annotator agreement and also to prevent errors and recognise difficult words.

In the following research, we want to make our method even more accurate by duplicating the batches once again, so every thousand words has been annotated by three different annotators. This could help us further explore the inter-annotator agreement and compare the two methods - two annotations vs. three of them.

3.2 Objective 2: Annotator recruitment

The annotation team consists of eight annotators, all of which are Czech native speakers and have finished their secondary education. They didn't receive full linguistic university education yet are relatively educated in the language area. This helps provide the sort of annotation data for later research use: the annotators don't assess the language too complexly, yet they do understand the subject enough so they can judge Czech headwords by their intuition.

Each annotator was asked about their local and social background – where they and their relatives live and lived and what schools and languages did they study. This information could be used later when examining the annotations separately.

4 Headword annotation phase

After preparations have been met, the project could step into the headword annotation phase which we called Phase 1.

4.1 Headword annotation

The headword annotation consists of a simple task of assigning a single *flag* to a potential headword. The annotator goes through a list of potential headwords (lemma-POS couples) and assigns a *flag* to each of them as follow:

1. If they don't understand the lemma, don't know it from the use of language or think it is not a proper word, the annotator is to choose the flag *I don't know*.
2. If they know the lemma from another language or assume it is used in another language, but don't know it from the use of Czech, the annotator is to choose the flag *not Czech*.
(Note: The flags *I don't know* and *not Czech* are handled very similarly in the proceeding phases.)
3. If the given lemma is a word in Czech (including non-lemma forms), but there is another word in standard contemporary Czech that is used much more often, the annotator is to choose the flag *non-standard*. Here, intuition of a common user of contemporary Czech should be preferred to the knowledge acquired in schools. Non-standard forms include the past, literary, dialectical, non-written and other word forms.
4. If the given lemma is a word form in standard contemporary Czech but it is not the lemma form, the annotator is to choose the flag *not a lemma*.
5. If the suggested lemma is a correct lemma form in standard contemporary Czech but the POS tag cannot be considered corresponding to the lemma, the annotator is to choose the flag *wrong part of speech*.

6. If the suggested headword contains a correct lemma form in standard contemporary Czech and the POS tag can be considered corresponding to the lemma, the annotator should choose whether the lemma is a *proper name* (flag) or not. For the proper nouns the flag **OK** is to be chosen.

4.2 Annotator training

Some additional training was needed so the annotators could understand their task. This required a work manual, a short introduction and presentation of the project in a workshop and also a discussion (brainstorming) about the language-related problems that can arise. Before, our limited knowledge about such problems in Czech was based on our language intuition and experience of the preceding dictionary-creation (Lao, Tagalog, Urdu and Ukrainian – see section 1). For annotators to understand the basic linguistic, language-neutral terminology used in our project, an interactive online course was provided. (For each phase, a course is needed. The course for headwords contained information about how to approach foreign words, non-standards words, proper names etc.)

A significant difference from the preceding projects is that the coordinators are newly also native speakers of the examined language (Czech) and can thus better comprehend the subject and anticipate difficulties.

4.3 Language-related annotation problems

Here are some of the language-related problems and solutions discussed on the training and during the annotation:

- **Only single words:** The batches contain only single words in combination of POS tags. This should be considered when we come across words of which their dictionary form usually includes another word. This in Czech mostly concerns the reflexive verbs (*reflexiva tantum*). For example the verb “bát se” doesn’t have an equivalent without the reflexive pronoun “se”. Yet the batches would only contain the headword “bát-verb” – this form should be accepted in spite of not having the obligatory pronoun.
- **Presumption of correctness:** POS tagging can be in some cases very complex. We encourage the annotators to accept the POS tag provided by the automatic tagger of the corpus. Only if the POS tag should be considered objectively wrong for certain, POS tag is not to be accepted. (E.g. the word “prostřednictvím-preposition” is to be considered OK, because it can behave like a preposition in this norm, even though it comes from the noun “prostřednictví”. On the other hand, “hajný-adjective” should be considered having a wrong POS – in spite of being derived from an adjective, it behaves only as a noun in modern Czech.) We also advise not to depend fully on the information learned in previous education but to follow the intuition of a native speaker and the knowledge of the language behaviour in general.

- **Abbreviations** have been decided to be handled as usual (single) words. This means their POS tag should correspond to their sentence usage. For example the abbreviation "dr." (doktor, doctor) is a noun, the abbreviation "např." (například, for example) is an adverb. The lemma of the abbreviations needs to have or lack a dot according to the used standard to be accepted ("cca" for circa without a dot, "např." for například with a dot).
- **Single letters** which do not stand alone as words (e.g. "ě" which is not a word in Czech or "A" for which the lemma "a" should be used) or standard used abbreviations (e.g. "r" – the proper form is "r." for rok) are not to be considered proper lemmas.
- **Vulgar** and otherwise taboo words should be looked upon as normal part of the lexicon and annotated as such.
- **Negation** of words: When should it and when should it not be accepted in the lemma? We decided not to accept negation in a lemma if the word is not considered negative tantum (doesn't have a non-negated form; e.g. "nenávidět") or secondary negative tantum (the negated form has a distinct meaning from a simple semantic negation of the non-negated variant). The annotators should always think about if the non-negated form is used (E.g. "neodmyslitelně" is used very often in Czech. The word "odmyslitelně" on the other hand is practically never used.)
- **Interjections**: Which forms should be accepted? We decided to only accept the most transparent forms (e.g. "kikirikí" or "kykyryký", but not "kykyrykýhyý"). While this could be considered a very subjective decision, we predict that in most cases there will be more and less transparent forms. As in the POS disambiguation annotation, we encourage the annotators to consider the lemma right if they don't consider the form strongly non-standard.
- Other wanted properties of a lemma were discussed, such as preserving the gender in the noun lemmas (i.e. "stolař" and "stolařka" should be considered two separate lemmas).

4.4 Findings

Before the annotations have begun, our vision was to firstly generate and annotate 15 000 potential words twice (approximately 10 000 future dictionary entries), possibly extending this number to 50 000 in the future. This estimation has been based on the speed of the previous projects. However, the annotations of Czech headwords were faster than previous annotations. One batch took a single person approximately 2 hours to annotate (meaning the double annotation took 4 hours), whereas a similar batch in Ukrainian took a single person 6 hours (12 for double annotation). One of the expected reasons is that Czech has a significantly better tools for Corpus creation and management (e.g. Majka

morphological analyser [5] and desamb [4]) and bigger corpora than the other languages. This also means more headwords get the flag *OK* (they contain the right lemma and POS tag) than in the previous projects. (For example, only 38.4 % of the Ukrainian headwords have been annotated as *OK*. [2] The same flag got 65.7 % of the Czech headwords in 149 batches completed to the day of writing this paper.)

We finally decided to extend the number of twice annotated headwords to 80 000.

We have chosen one batch that every annotator should annotate that could provide us some interesting data. This data have been used to recognise the annotation style of each annotator and recognise some interesting linguistic problems.

As mentioned before, other batches have been annotated by two different annotators. We are considering annotating these for a third time since annotations since the speed of annotations is higher than expected. Before this, experimental third annotation of one or two batches will be made and we will examine the statistics provided by these experiments.

5 Subsequent phases

In this section, two of the nearest subsequent phases are described. The tasks follow on from the annotation data provided in Phase 1 and the headwords lists generated in Phase 0. Both phases are going to be launched simultaneously, but they could also follow each other if needed in other projects.

5.1 Revision phase

Revisions of the headwords annotations (also called Phase 2) are done by experienced annotators who proved capable in Phase 1. We have chosen 4 annotators who worked the longest and we have examined their annotation data. In the data of every annotator, we spotted recurring difficulties. These will be discussed on an upcoming training for revisions.

The task of revisions is to go through headwords at least once annotated *non-standard*, *not a lemma* or *wrong POS*. Headwords with the *I don't know* or *not Czech* flag in combination with the *proper name* or *OK* flag are also to be revised. The same goes for the headwords with the combination of the *proper name* and *OK* flag. The revising annotator sees a headword and its annotation flag and is supposed to select one of these options: Either they can enter the correct headword the annotated headword corresponds to. Or they can state they don't understand the word or the word is not Czech. The last option is to state the annotated headword is correct.

The headwords annotated only as *OK* or only as *proper name* are not revised. The same goes for headwords annotated only in any combination of the *I don't know* and *not Czech* flags. This focuses Phase 2 only on a fraction of the headword list with the need for revisions. As mentioned in subsection 4.4, the number

of the headwords annotated with the OK flag is greater than in the previous projects since the Czech tagger is more precise and the used corpora are bigger than for the languages before. The speed of revisions can thus be also expected to increase.

5.2 Forms

Another aspect of the dictionary we want to create, besides the lemmas and POS tags, are the inflected word forms (we call this task Phase 3). In Czech, nouns, adjectives, pronouns, numerals and verbs can be inflected and adverbs can be comparative. The form annotators will go through a list of headwords who have been decided to be standard Czech lemmas with corresponding POS tags in Phase 1 and later Phase 2 (section 4 and subsection 5.1). For each headword, a list of possible word forms will be generated from the corpus. The task is to mark which of them are correct standard forms of the given headword.

6 Conclusion

This paper introduces the project of creating an express Czech dictionary using a semi-automatic method. In the first section, we describe the goals and priorities of the project (mainly the speed of creation) and the preparations needed to set up the creation. First, a list of headwords (lemma-POS pairs) are created from large corpora using automatic tools. In the following sections, first three phases of the project are introduced. First phase focuses on headwords: whether each lemma is correct and used in standard Czech and whether the POS tag corresponds with it. Second and third phase follow the first phase. The second phase focuses on revising the headwords that in the first phase have not been annotated as completely correct or completely incorrect. In the third phase, annotators decide which automatically found words are correct inflected forms of a headword.

After the first three phases, more automatic and manual tasks are going to follow. The most demanding, time-heavy phases are estimated to be the ones focused on meaning distinction: sense-recognising, thesaurus words and examples. [2]

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: Proceedings of the 6th Biennial Conference on Electronic Lexicography. pp. 805–818. Lexical Computing CZ s.r.o., Brno, Czech Republic (2019), https://elex.link/elex2019/wp-content/uploads/2019/10/eLex-2019_Proceedings.pdf

2. Blahuš, M., Cukr, M., Herman, O., Jakubíček, M., Kovář, V., Kraus, J., Medveď, M., Ohlídálová, V.: Rapid Ukrainian-English Dictionary Creation Using Post-edited Corpus Data. In: Medveď, M., Měchura, M., Tiberius, C., Kosem, I., Kallas, J., Jakubíček, M., Krek, S. (eds.) *Electronic lexicography in the 21st century (eLex 2023): Invisible Lexicography*. Proceedings of the eLex 2023 conference. Lexical Computing CZ s.r.o., Brno, Czech Republic (2023), <https://ellex.link/ellex2023/wp-content/uploads/114.pdf>
3. Jakubíček, M., Kilgarrieff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family (2013), <https://api.semanticscholar.org/CorpusID:107998183>
4. Šmerk, P.: K morfologické desambiguaci češtiny [online] (2008 [cit 2023-11-07]), <https://is.muni.cz/th/wteg5/>
5. Šmerk, P.: Fast Morphological Analysis of Czech. In: *Proceedings of the Raslan Workshop 2009*. Masarykova univerzita, Brno (2009), <https://nlp.fi.muni.cz/raslan/2009/papers/13.pdf>

Development of the NVH Schema Format for Lexicographic Purposes

Marek Medved^{1,2}, Miloš Jakubíček^{1,2}, Vojtěch Kovář^{1,2}, and Tomáš Svoboda¹

¹ Lexical Computing,
Brno, Czechia

² Masaryk University
Brno, Czechia

`name.surname@sketchengine.eu`

Abstract. A unified e-dictionary entry format is one of the most important things to consider when building a new dictionary. In the Lexonomy tool, where the new NVH lightweight markup language is used to store dictionary data, an NVH schema is assigned to each dictionary, specifying the NVH structure belonging to each dictionary entry. Until now, the schemata used in Lexonomy were quite limited and focused only on the position of a node in the NVH hierarchy and on the arity of its occurrence. In the recent development, we identified a need for a more fine-grained restriction mechanism and, therefore, extended the NVH schema format so that it can also inspect the value of each node according to its type and confirm match according to a predefined regular expression.

Keywords: NVH, XML, Name-Value Hierarchy, Lexonomy, Sketch Engine

1 Introduction

The Name-Value Hierarchy or NVH is a lightweight markup language targeted at dictionary development [1]. It is a user-friendly alternative to XML-encoded plain text formats, currently used in many digital dictionaries, and store dictionary entries in a more compact and readable form.

The NVH language is currently the backbone format of the Lexonomy system [2,3], which is focused on digital dictionary development. Recent developments of the Lexonomy system led to changes that needed to be incorporated into the NVH language, especially the schema that specifies the final structure of the developed dictionary.

In this paper, we will introduce dictionary schema modifications that restrict each node value inside the NVH according to the model needs.

2 Name-Value Hierarchy markup language

Name-Value Hierarchy (NVH) is a plain text-based format similar to XML with much simpler syntax. An NVH file consists of a list of nodes, where each node

has its name and optional value separated by a colon-space character. Each node can also contain a list of child nodes prefixed by Python-like indentation (see Figure 1).

```
hw: car
  lemma: car
  pos: NOUN
  image: car.jpg
    quality: good
    explicit: no
```

Fig. 1: NVH sample

3 Schema

A dictionary schema is always required along with the dictionary content in NVH. The purpose of the schema is to avoid nodes that are not important or unwanted for the dictionary in hand and also to automatically check if the dictionary content complies with the predefined dictionary schema.

The previous version of an NVH schema supported essential value restriction that defines the number of required nodes with the specific name (see Figure 2). Using the question mark character ("?",) on the position of the value, the schema allows the use of none or one node with the specified name (i.e., *lempos*). The plus character ("+",) requires at least one node with the name inside the dictionary (like *hw*). Similarly, the number followed by the plus character requires using at least the given number of nodes. The star character ("*",) puts no restrictions on the node as it can appear from zero to infinity times (i.e., *audio*). Finally, the names with no value in the schema must appear exactly once (i.e., *lemma*).

The expressive ability of the previous schema implementation was very limited, and could not catch most mistakes made by annotators. For example, we could require *lempos* attribute to be present in each dictionary entry. Still, there is no option to set the format of the *lempos* value if we want it to be a combination of the lemma and one character representing the part of speech. Therefore, we expanded the original schema form with a new set of parameters.

In this new NVH schema, we introduce an extended definition for the number of nodes, type of the value, and usage of regular expressions for string values.

3.1 Extended definition for number of nodes

Similarly to the previous version, the new schema definition can encode how many nodes with the specific name are required inside the dictionary entry.

```

hw: +
  lemma:
  lempos: ?
  pos:
  freq: ?
  audio: *
  image: 2+
    quality: ?
    explicit: ?
    source: ?
  examples:
    example: 2+
  translation: *
    language:
  affiliation: ?

```

Fig. 2: Basic NVH schema node restrictions

Using the “?”, +, *, and 2+, we can restrict the number of nodes as we introduced above. On top of that, the new schema definition introduces range “1-5” that can set the upper bound as well as the lower bound, which was not previously available (i.e., image node in Figure 3).

```

hw: +
  lemma: ?
  lempos: ? ~.*-.
  pos: ?
  freq: ? int
  audio: * audio
  image: 1-5 image
    quality: ? ["good","bad"]
    explicit: ? bool
    source: ? url ~.*pixabay.*
  examples: empty
    example: 2+ ~.{1,50}
  translation: *
    language: ~.{3}
  affiliation: * ["MU (Brno)", "VUT \"Brno\"", "UK, Praha"]

```

Fig. 3: NVH schema with all supported value restrictions

3.2 Value types

The new schema format introduces the support for typing. There are seven types that have been developed according to Lexonomy usage and should cover the majority of user needs inside the Lexonomy tool:

- audio type is a string that restricts the node value an audio file by matching the string with a list of supported audio extensions. In the current version we support these audio extensions: .3gp, .aa, .aac, .aax, .act, .aiff, .alac, .amr, .ape, .au, .awb, .dss, .dvf, .flac, .gsm, .iklax, .ivs, .m4a, .m4b, .m4p, .mmf, .movpkg, .mp3, .mpc, .msv, .nmf, .ogg, .oga, .mogg, .opus, .ra, .rm, .raw, .rf64, .sln, .tta, .voc, .vox, .wav, .wma, .wv, .webm, .8svx, .cda.
- bool type restrict the node value to just Boolean values True and False. These two values can be expressed by: True, False, true, false, Yes, No, yes, no, 0, 1.
- empty type does not put any restriction on the value but requires the value to be empty. This type should be used for the nodes that introduce a container, like *examples* node in Figure 3.
- image type is the same as audio type except for the list of supported extensions: .jpeg, .jpg, .png, .gif, .bmp, .tiff, .svg, .raw, .ico, .webp, .heic, .heif, .psd, .eps, .ai, .tga, .pdf.
- int type restricts the value only to integer numbers.
- list type is not explicitly used inside the NVH schema but is determined according to the list of possible values (like *quality* and *affiliation* in Figure 3).
- string is the default type and does not need to be explicitly used in the schema.
- url type confirms if the node's value is a URL link.

3.3 Regular expressions

Character-based types url and string additionally support regular expression restrictions that have to match with the node value. The tilde character (~) always introduces a regular expression. The format of the regular expression follows the Python regular expression syntax of the re module³ or any other user-specified format that should be provided as a comment in the schema (a comment is any line starting with the # character).

3.4 NVH script modifications

The Python script nvh.py is adopted to the above-mentioned modifications. The schema validation, as well as schema generation operations, are modified to account for the new types and regular expressions.

³ <https://docs.python.org/3/library/re.html>


```

{
  "hw": {"min": 1, "max": Infinity, "type": "string",
    "children": ["lemma", "lempos", "pos", "freq", "audio", "image",
      "examples", "translation", "affiliation"]},
  "lemma": {"max": 1, "type": "string"},
  "lempos": {"max": 1, "type": "string", "re": ".*-."},
  "pos": {"max": 1, "type": "string"},
  "freq": {"max": 1, "type": "int"},
  "audio": {"max": Infinity, "type": "audio"},
  "image": {"children": ["quality", "explicit", "source"],
    "min": 1, "max": 5, "type": "image"},
  "quality": {"max": 1, "type": "list", "values": ["good", "bad"]},
  "explicit": {"max": 1, "type": "bool"},
  "source": {"max": 1, "type": "url", "re": ".*pixabay.*"},
  "examples": {"children": ["example"], "min": 1, "max": 1, "type": "empty"},
  "example": {"min": 2, "max": Infinity, "type": "string", "re": ".{1,50}"},
  "translation": {"children": ["language"], "max": Infinity, "min": 0,
    "type": "string"},
  "language": {"min": 1, "max": 1, "type": "string", "re": ".{3}"},
  "affiliation": {"max": Infinity, "min": 0, "type": "list",
    "values": ["MU (Brno)", "VUT \\\"Brno\\\"", "UK, Praha"]}
}

```

Fig. 4: JSON export of the NVH schema from Figure 3

For Lexonomy purposes, we also include a new type of export. The NVH schema can now be exported into the JSON format that can be useful for any tool incorporating the NVH format. Currently, this export is used by the Lexonomy system frontend to validate whether annotators' input follow the restrictions defined by the schema before being stored in the final dictionary NVH file. Figure 3 presents an example of JSON export of the schema from Figure 4.

4 Conclusions

This paper presents new NVH schema modifications that allow dictionary managers to define more precisely how the dictionary entry will be developed. The new type of restrictions together with regular expressions can exactly specify the value of each node. This unique design of the NVH schema strictly directs the annotators during the dictionary development and avoids mistakes and unnecessary post-processing of inconsistent annotations.

References

1. Jakubíček, M., Kovář, V., Měchura, M., Rambousek, A.: Using NVH as a Backbone Format in the Lexonomy Dictionary Editor. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2022. (2022) 55–61

2. Měchura, M.B., et al.: Introducing Lexonomy: an open-source dictionary writing and publishing system. In: *Electronic Lexicography in the 21st Century: Lexicography from Scratch*. Proceedings of the eLex 2017 conference. (2017) 19–21
3. Rambousek, A., Jakubíček, M., Kosem, I.: New Developments in Lexonomy. *Electronic lexicography in the 21st Century (eLex 2021) Post-editing lexicography (2021)* 86

Data Gathered with Automatic Tools from European Parliamentary Chambers

Ota Mikušek^{1,2}

¹ Faculty of Informatics, Masaryk University
xmikusek@fi.muni.cz

² Lexical Computing, Brno, Czech Republic
ota.mikusek@sketchengine.eu

Abstract. This paper reflects on the set of tools developed in my bachelor's thesis, titled "Continuous Automatic Development of European Parliamentary Corpora." Despite the existence of numerous corpora offering speeches from the parliaments of the European Union, the developed toolset is designed to gather and build such corpora with minimal human intervention. With nine months of practical application, this paper presents insights into the faced challenges and their respective solutions, providing an overview since the initial release of the toolset.

Keywords: parliamentary protocols, continuous downloading, corpus processing, automatic tools, corpus development, automatic maintenance of tools

1 European Parliamentary Corpora

Between July 2020 and May 2021, the ParlaMint I [4] project aimed to create corpora of transcriptions from the sessions of 17 European Union parliaments from 2015 to October 2019. ParlaMint I was the largest project of its kind for European parliamentary corpora at the time. Each parliamentary corpus had a dedicated lead developer, which helped the overall quality of the resulting corpora.

In December 2021, the ParlaMint II [3] project extended the work of ParlaMint I by including parliamentary transcriptions up to July 2022. This project also involved updates to the schema, validation, and enhancement of corpora with additional metadata.

In July 2023 ParlaMint 3.0 [2] began as a follow-up to ParlaMint II. ParlaMint 3.0 added new metadata information for bicameral parliaments if data was provided from the upper or the lower house of parliament. New corpora were introduced in ParlaMint 3.0, namely corpora of Austria, Bosnia, Catalonia, Galicia, Greece, Norway, Portugal, Serbia, Sweden, and Ukraine. Two corpora (Spanish and Lithuanian) were removed.

The ParlaMint projects provide unified metadata for all corpora, consisting of 24 types of information, including timestamps, speaker details, transcriber

notes, and source URLs for documents. However, it's important to note that despite their rich metadata, only 19 out of the 27 current EU states are covered by ParlaMint. Expanding coverage to include these missing parliaments is a future objective for the ParlaMint project.

In addition, there are other initiatives to create parliamentary corpora, such as the Polish Parliamentary Corpus [7], which covers debates from 1919 to the present, and the German Parliamentary Corpus (GerParCor) [1], which includes transcripts from Germany, Liechtenstein, Austria, and Switzerland up to 2021, with plans for continuous development. The Czech Parliamentary Corpus (CzechParl) [5] is based on Czech parliament stenographic protocols from the 1990s. The Dutch Parliamentary Corpus (DutchParl) [6] aims to collect Dutch parliamentary documents and has different sized corpora for Belgium, Flanders, and the Netherlands, with ongoing development efforts.

2 Automatic tools

The outcome of my thesis, titled "Continuous Automatic Development of European Parliamentary Corpora," is a Python-based toolset designed to facilitate the ongoing automatic development of corpora derived from transcriptions of parliamentary sessions involving selected members of the European Union. The toolset employs scripts that gather protocols from suitable sources on chamber websites, accommodating various formats and unifying them into a standardized prevertical format. The prevertical³ format is a file format containing plain text and structures. The structures enclose the text and provide metadata about the text.

The scripts are designed to operate independently of each other, functioning autonomously, automatically, and atomically. Each script comprises three main components: shared code, a tool for discovering and downloading new protocols, and a tool for processing the downloaded protocols into prevertical files. In the event of an error, the scripts have the capability to log the error, notify the script administrator, and revert to the last consistent state.

The source code of all the tools is licensed under GNU Lesser General Public License 3.0 and available in a GitLab repository.⁴

2.1 Downloading of data

To secure reliable sources of protocols, a search was conducted on official parliamentary websites. To be deemed reliable, a source must originate directly from the parliament, offer a mechanism to identify newly added protocols, and refrain from dependence on website-provided scripts, particularly those depending on JavaScript.

The reason why script execution to access or discover new protocols is unwanted is that user-side scripts can change over time, and these changes

³ https://www.sketchengine.eu/my_keywords/prevertical/

⁴ <https://gitlab.com/Atom194/european-parliamentary-protocols>

may cause errors during the automatic download process. Such dependency is unwanted because it increases maintenance difficulty.

The identified sources presented data in various formats, including plain text, HTML, JSON, CSV, XML, XLSX, and DOCX. Additionally, some of the chambers provided PDF files with transcriptions. However, challenges arose with the PDF format, specifically regarding the ordering of paragraphs and text extraction, especially when words were hyphenated at the end of a line using the “-” character. In instances where the source was not available on the parliament website, the parliament was connected through email.

The developed scripts automatically and atomically download protocols from designated sources. In the event of a protocol download failure, the error information is logged, and the download will be retried during the next script execution.

2.2 Processing of protocols

A script that processes downloaded protocols called *prevertbuilder* was created for each chamber website. The *prevertbuilder* is responsible for metadata extraction and unifying downloaded protocols into *prevertical* format.

The *prevertbuilder* works like a pipe. It contains the initialization, writing, and finalization methods, which process downloaded protocols linearly and do not require the whole protocol to be loaded in memory. This capability is used, for example, in the Swedish parliament, where one downloaded document consists of protocols from a month period.

A protocol is marked as successfully processed only when *prevertbuilder* process the protocol without an error. *Prevertbuilders* are capable of detecting presence of new information (for example, new tags or attributes) in processed protocols. By default, in these cases, protocols are processed without these new elements. However their occurrence is logged as a warning in the script log.

3 Tools maintenance

During the continuous nine-month operation, the tools underwent several modifications to accommodate changes in the source data. These adjustments primarily focused on adapting the *prevertical* creation process to handle new elements, structures, and attributes in the sources.

For instance, a change emerged within Slovenia’s parliament, where changes in month naming conventions were made after the first tool deployment. The updated month names differ from the previous ones in inflection of the month names. The solution to this change involved adding records to the month name to month number dictionary as errors arose from unknown month names. Due to a lack of knowledge in Slovenian inflection, this approach proved more manageable than attempting to add all new month names simultaneously, as errors were prone to occur in that process.

⁵ The chamber releases new transcriptions yearly.

⁶ The chamber releases new transcriptions yearly.

Table 1: Comparison of processed data from May 2023 to November 2023

corpus name	words	words now	change	from year
bg_deputies	5.40M	5.82M	+0.42M	2022
cz_deputies	18.41M	20.71M	+2.30M	2018
cz_senate	11.32M	11.51M	0.19M	2010
dk_deputies	79.00M	79.55M	+0.55M	2007
nl_deputies	71.20M	80.20M	+9.00M	2013
nl_senate	9.99M	11.01M	+0.02M	2019
ir_deputies	40.70M	87.28M	+46.58M	2022
ee_deputies	9.04M	10.47M	+1.43M	2020
fi_deputies	21.09M	21.09M	0 ⁵	2015
be_deputies	54.94M	56.70M	+1.76M	2007
be_senate	0.06M	0.69M	+0.63M	2019
fr_deputies	21.09M	59.55M	+38.46M	2015
fr_senate	169.08M	173.52M	+4.44M	2004
at_deputies	6.94M	7.19M	+0.25M	2022
at_senate	2.73M	2.87M	+0.14M	2019
de_deputies	125.03M	125.53M	+0.50M	1950
gr_deputies	58.31M	59.47M	+1.16M	2015
hu_deputies	3.08M	3.93M	+0.85M	2022
it_deputies	3.32M	5.15M	+1.83M	2022
it_senate	13.31M	14.61M	+1.30M	2018
pl_senate	20.08M	20.25M	+0.17M	2011
pt_deputies	141.10M	154.36M	+13.26M	1976
ro_deputies	14.02M	14.86M	+0.84M	2016
ro_senate	26.36M	26.88M	+0.52M	2001
sk_deputies	6.76M	8.73M	+1.97M	2022
si_deputies	15.49M	23.69M	+8.20M	2018
es_deputies	66.66M	68.73M	+2.07M	2019
se_deputies	131.74M	131.74M	0 ⁶	1994
sum	1,146.25M	1,286.09M	+139.84M	-

Changes also happened in the Parliament of Bulgaria, which implemented specific measures to block requests not containing a 'User-Agent' header. This change caused the tool to be unable to download any protocol. The tool was modified to use 'User-Agent': 'curl/7.82.0' header, which resolved the problem.

Sometimes, when a protocol is being downloaded, the connection fails, and the tool ends up in an error state. This is the most common type of error in the toolset. Out of 299 errors encountered during past nine months, 72 were caused by connection failure. The tools feature robust error recovery mechanisms, allowing them to seamlessly roll back to the last stable state in the event of any encountered errors. In such cases, the problematic protocol is automatically reattempted for download during the subsequent execution of the tool.

4 Gathered data

The resulting preverticals underwent a thorough error check. Corpora were then generated from all preverticals, and an analysis was conducted on the top 100 keywords, as well as the most frequently occurring 500 words in each corpus. This analysis aimed to identify any potential presence of source metadata that might not be part of the protocol text.

As of now, the entire toolset has compiled a total of 1,286.09 million words sourced from 28 chambers within the EU parliaments, out of the 38 chambers available. This collection spans across 17 languages, namely Bulgarian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Hungarian, Italian, Portuguese, Romanian, Slovak, Slovenian, Spanish, and Swedish. Statistics of each parliamentary chamber can be found in Table 1.

One notable property of some chambers is their grammatical correctness in transcriptions, even though the speaker does not speak grammatically correctly. Therefore, the gathered data are also grammatically correct. This property can be found in chambers such as chambers of the Czech Republic, Slovakia, Ireland, and possibly others, depending on the internal policy of the chamber.

For instance, in the lower chamber of the Czech Republic, transcriptions are transcribed into grammatically correct language, even though transcribed speech contains ungrammatical language. An exception is made for instances of a speaker delivering a strongly emotional speech. Corrections are applied in cases involving incorrect endings or inflection, addressing obvious errors in verbosity, stuttering in speech, and similar linguistic inaccuracies. Obvious mispronunciations are corrected, unless subsequently addressed in the following speeches. Corrections also include addressing the excessive use of personal and demonstrative pronouns, as well as repetition of words, unless such repetition serves an emphatic purpose. It is important to note that there are no corrections made for factual errors or instances of offensive or obscene language.

5 Conclusions

The size of gathered data is continuously growing. In addition to collecting textual data, these tools gather metadata associated with the texts. Common metadata across all sources include the names of the speaker and the date of the speech. Additional metadata is provided for specific chambers, such as notes from the transcriber, party affiliation, the role of the speaker in the chamber, and other relevant details.

However, it is crucial to acknowledge that the quality of the extracted metadata depends on the quality and formatting of the source. Consequently, errors may occur in both the metadata and texts due to the inability to autonomously distinguish between text and metadata in the source. For example, some of the older transcriptions of the German parliament were gathered by OCR, and the resulting scans are sometimes missing a separator of speaker and speech. In the Romania upper chamber of parliament, the role and name of the speaker are sometimes used as the name of the speaker.

References

1. Abrami, G., Bagci, M., Hammerla, L., Mehler, A.: German parliamentary corpus (gerparcor). In: Proceedings of the Language Resources and Evaluation Conference. pp. 1900–1906. European Language Resources Association, Marseille, France (June 2022), <https://aclanthology.org/2022.lrec-1.202>
2. Erjavec, T., Kopp, M., Ogrodniczuk, M., Osenova, P., Fišer, D., Pirker, H., Wissik, T., Schopper, D., Kirnbauer, M., Mochtak, M., Ljubešić, N., Rupnik, P., Pol, H.v.d., Depoorter, G., de Does, J., Simov, K., Grigorova, V., Grigorov, I., Jongejan, B., Haltrup Hansen, D., Navarretta, C., Mölder, M., Kahusk, N., Vider, K., Bel, N., Antiba-Cartazo, I., Pisani, M., Zevallos, R., Regueira, X.L., Vladu, A.I., Magariños, C., Bardanca, D., Barcala, M., Garcia, M., Pérez Lago, M., García Louzao, P., Vivel Couso, A., Vázquez Abuín, M., García Díaz, N., Vidal Miguéns, A., Fernández Rei, E., Diwersy, S., Luxardo, G., Coole, M., Rayson, P., Nwadukwe, A., Gkoumas, D., Papavassiliou, V., Prokopidis, P., Gavrilidou, M., Piperidis, S., Ligeti-Nagy, N., Jelencsik-Mátyus, K., Varga, Z., Dodé, R., Barkarson, S., Agnoloni, T., Bartolini, R., Frontini, F., Montemagni, S., Quochi, V., Venturi, G., Ruisi, M., Marchetti, C., Battistoni, R., Dargis, R., van Heusden, R., Marx, M., Depuydt, K., Tunland, L.M., Rudolf, M., Nitoń, B., Aires, J., Mendes, A., Cardoso, A., Pereira, R., Yrjänäinen, V., Norén, F.M., Magnusson, M., Jarlbrink, J., Meden, K., Pančur, A., Ojsteršek, M., Çöltekin, Ç., Kryvenko, A.: Linguistically annotated multilingual comparable corpora of parliamentary debates ParlaMint.ana 3.0 (2023), <http://hdl.handle.net/11356/1488>, slovenian language resource repository CLARIN.SI
3. Erjavec, T., Ogrodniczuk, M., Osenova, P., Ljubešić, N., Simov, K., Grigorova, V., Rudolf, M., Pančur, A., Kopp, M., Barkarson, S., Steingrímsson, S., van der Pol, H., Depoorter, G., de Does, J., Jongejan, B., Haltrup Hansen, D., Navarretta, C., Calzada Pérez, M., de Macedo, L.D., van Heusden, R., Marx, M., Çöltekin, Ç., Coole, M., Agnoloni, T., Frontini, F., Montemagni, S., Quochi, V., Venturi, G., Ruisi, M., Marchetti, C., Battistoni, R., Sebők, M., Ring, O., Dargis, R., Utka, A., Petkevičius, M., Briedienė, M., Krilavičius, T., Morkevičius, V., Bartolini, R., Cimino, A., Diwersy, S., Luxardo, G., Rayson, P.: Linguistically annotated multilingual comparable corpora of parliamentary debates ParlaMint.ana 2.1 (2021), <http://hdl.handle.net/11356/1431>, slovenian language resource repository CLARIN.SI
4. Erjavec, T., Ogrodniczuk, M., Osenova, P., Ljubešić, N., Simov, K., Pančur, A., Rudolf, M., Kopp, M., Barkarson, S., Steingrímsson, S., Çöltekin, Ç., de Does, J., Depuydt, K., Agnoloni, T., Venturi, G., Pérez, M.C., de Macedo, L.D., Navarretta, C., Luxardo, G., Coole, M., Rayson, P., Morkevičius, V., Krilavičius, T., Dargis, R., Ring, O., van Heusden, R., Marx, M., Fišer, D.: The parlamint corpora of parliamentary proceedings. Language Resources and Evaluation (Feb 2022). <https://doi.org/10.1007/s10579-021-09574-0>, <https://doi.org/10.1007/s10579-021-09574-0>
5. Jakubíček, M., Kovář, V.: Czechparl: Corpus of stenographic protocols from czech parliament. In: Proceedings of Recent Advances in Slavonic Natural Language Processing 2010. pp. 41–46. Masaryk University, Brno (2010)
6. Marx, M., Schuth, A., et al.: Dutchparl. a corpus of parliamentary documents in dutch. Proceedings Language Resources and Evaluation (LREC) pp. 3670–3677 (2010), https://pure.uva.nl/ws/files/990556/88437_332665.pdf
7. Ogrodniczuk, M.: Polish parliamentary corpus (2018), <http://hdl.handle.net/11321/467>, CLARIN-PL digital repository

Towards Perfection of Machine Learning of Competing Patterns

The Use Case of Czechoslovak Patterns Development

Ondřej Sojka  and Petr Sojka 

Faculty of Informatics, Masaryk University, Brno, Czech Republic
454904@mail.muni.cz, sojka@fi.muni.cz

Abstract. Finding space- and time-effective even *perfect* solution to the dictionary problem is an important practical and research problem, which solving may lead to a breakthrough in computation. Competing pattern technology from T_EX is a special case, where for a given dictionary a word segmentation is stored in the competing patterns yet with very good generalization quality. Recently, the unreasonable effectiveness of pattern generation has been shown – it is possible to use hyphenation patterns to solve the dictionary problem jointly even for several languages without compromise.

In this article, we study the effectiveness of *patgen* for the supervised machine learning of the generation of the Czechoslovak hyphenation patterns. We show the machine learning techniques to develop competing patterns that are close to being perfect. We evaluate the new approach by improvements and space savings we gained during the development and finetuning of Czechoslovak hyphenation patterns.

Keywords: dictionary problem, effectiveness, hyphenation patterns, *patgen*, syllabification, Czech, Slovak, Czechoslovak patterns, machine learning

“When you’re passionate about something, you want it to be all it can be.”
Debra Messing

1 Introduction

Dictionary problem is the task of storing a dictionary seen as a database of words where we distinguish the key part (the word) and the data part (values of the key). Finding space- and time-effective even *optimal* solution to the dictionary problem is an important practical and research problem. Solving it may lead to a breakthrough in computation. The effectiveness of the solution lies in the implicit data structures used. Typically some sort of trees (B-trees [1], tries) or hashing or their combination is used [4]. Time complexity is constant

$O(1)$ for both tree-based solutions (constant C is the tree depth to locate values in the list or hash computation time) and space in $O(D)$, e.g. linear in dictionary storage size D . Absolute value of C and linear coefficient for D are important.

In T_EX, a solution to the dictionary problem is used for hyphenation. For a given *key*, e.g. a word to be hyphenated, the *values* are the positions of a word where hyphenation may occur. To minimize the storage size of ever-growing dictionaries Frank Liang designed the *competing pattern* technology for T_EX [6]. The dictionary problem is decomposed in such a way that word segmentation is stored in the competing patterns generated from the already hyphenated wordlist.

Recently, the unreasonable effectiveness of pattern generation [10] has been shown. It is possible to use hyphenation patterns to solve the dictionary problem even for several languages without compromise. Also, multiple languages could be covered in the same set of patterns [12,7]. All these developments trigger the necessity of effectiveness and of bringing new solutions.

In this article, we show the effectiveness of *patgen* for the generation of the Czechoslovak hyphenation patterns that are close to being optimal.

The paper is structured as follows. We describe competing patterns in Section 2. We define pattern development processes in machine learning nomenclature and define metrics for rigorous evaluation of dictionary problem solutions by competing patterns in Section 3. Section 4 shows an experiment with the hyphenation model development and dataset cleaning. Experiment with grid search of parameter generation and the achieved results are in Section 5. We show that designed techniques and the grid search of parameter generation lead to the development of hyphenation patterns with effectiveness improvements and space savings on the use case of Czechoslovak hyphenation patterns.

Finally, we describe the potential for future work in Section 6 and conclude by Section 7.

“All fixed set patterns are incapable of adaptability or pliability.
The truth is outside of all fixed patterns.” – Bruce Lee

2 Competing Patterns

Frank Liang [6] designed an efficient solution to a dictionary problem with *competing patterns*. Patterns are *generated* from the dictionary in the form of an already hyphenated wordlist with program *patgen*. [3]

Generation is decomposed into phases called levels. In each level, all character patterns in the range of length are considered. Patterns added in odd levels are covering, they add new hyphenation points given the letter context, while in even levels and inhibiting, e.g. forbid hyphenation points. Iteration of covering and inhibiting levels creates a hierarchy of exceptions. The patterns generated in odd levels *compete* with those generated in even levels whether to hyphenate or not.

The key to having both high-coverage and small sets of patterns with no bad hyphenation point allowed lies in the setting of thresholds for each level

that decide whether patterns will or will not be included in the final set of patterns. [8]

An example of competing patterns generation from the Czechoslovak wordlist of cca 600,000 hyphenated words (8.5 MB) is in [11, Table 2]. The generated pattern dictionary of 8,231 patterns has a size of 45 kB. Patterns loaded into RAM in the packed trie data structure are even smaller, reaching a compression ratio of around 2000:1. The hyphenation value for the input word is found in the constant time of several instructions needed to reach the list of trie storing the pattern.

The competing pattern generation technique thus maps the dictionary problem of storing the hyphenation point for all words of language into the dictionary problem of storing orders of magnitude smaller sets of short patterns.

Another *crucial* advantage of pattern-based solution is that short patterns learn *hyphenation rules* that are applicable to words not seen during training. As new words steadily appear in natural languages, learning hyphenation rules rather than hyphenated wordlist brings new *generalization* properties.

“In God we trust, all others bring data.” — W Edwards Deming

3 Evaluation Metrics

The preparation of patterns from a wordlist is a typical *supervised machine learning* solution to dictionary problems.

There are four numbers in the confusion matrix (also called contingency table) that compare hyphenation point prediction by patterns with the ground truth expressed in the wordlist: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In the evaluation results, we report several metrics:

Good sum or percentage of found hyphenation points as a TP,

Bad sum or percentage of badly suggested hyphenation points (FP, type 1 error),

Missed sum or percentage of missed hyphenation points (FN, type 2 error),

Precision defined as $\frac{\text{Good}}{\text{Good}+\text{Bad}} = \frac{\text{TP}}{\text{TP}+\text{FP}}$,

Recall defined as $\frac{\text{Good}}{\text{Good}+\text{Missed}} = \frac{\text{TP}}{\text{TP}+\text{FN}}$,

F-score, F_β defined as $F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} = \frac{(1 + \beta^2) \cdot \text{TP}}{(1 + \beta^2) \cdot \text{TP} + \beta^2 \cdot \text{FN} + \text{FP}}$,
where a positive real factor β is chosen such that **Recall** is considered β times as important as **Precision**.

As **Precision** is much more important than **Recall**, we report $F_{1/7}$ and $F_{1/77}$: type 1 errors are more severe than type 2 errors in our hyphenation points setup.

Nonzero **Bad** or **Missed** results do not necessarily mean that the patterns performed badly, the opposite is often the case – patterns have found a rule that is not obeyed in the ground truth wordlist. In other words, the patterns found

an inconsistency that needs to be fixed in the underlying wordlist, rather than a valid exception.

There are two main parts of the machine learning (ML) solution: *model development* and *model evaluation*. The practice of manually inspecting and fixing bad hyphenation points has been used during the *model development* of the wordlist so that the data do not contradict each other. **Precision**, sometimes called *coverage*, tells how many hyphenation points used in training were correctly predicted by the patterns.

The *model evaluation* of the quality of developed patterns could be done with the same metrics as for the model development of a hyphenated wordlist.

Evaluation of *generalization* properties, e.g. how the patterns behave on unseen data, has to be done on the words *not* available in the data used during patgen training. The dataset has to be split into non-overlapping training and validation test sets.

To assess the *generalization* properties, we used 10-fold *cross-validation*, leaving *validation dataset* – one-tenth out of the training set – to evaluate the effectiveness metrics of the patterns on unseen words.

There are other effectiveness metrics that could be measured in the dictionary task:

speed of getting values for the given key (word), and
size size of data structure to store keys (words, patterns).

The above metrics computed for our use case of Czechoslovak patterns are reported in tables 2 and 3. It is clear that when adding 33 bad hyphenation points as full word patterns, the coverage is 99.99% with no error on seen words and only 0.15% error rate on unseen new words.

It has been proven that the task of creating the minimal pattern set is NP-complete [9].

“Pleasure in the job puts perfection in the work.” – Aristotle

4 Dataset Consistency for Model Development

Even though the previous results testify to unreasonable effectiveness [10], we have designed a model development task by improving consistency of syllable markup. The rationale is that when inconsistent hyphenation points are marked in the data, more patterns are needed to cover all those idiosyncrasies.

Natural language is in continual development. In Czech and Slovak, some compound words like *roz-um* are no longer considered compounds with hyphenation points separating constituent words. Instead, syllabic hyphenation *ro-zum* is preferred.

Further, syllabic rules hold also near the word border, while it is forbidden to hyphenate so that a single character is cut during hyphenation.

We have semiautomatically filtered 25,273 words that start with one character vowel syllable (aeiouy), and added a hyphenation point after it in patgen

Table 1: Pattern generation parameters: statistics from the generation of Czechoslovak hyphenation patterns in 2020 [11] with correct optimized patgen generation parameters (correctopt2020)

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	2,032	2,800,136	242,962	55,605	1 3	1 5 1
2	2,009	2,791,326	10,343	64,415	1 3	1 5 1
3	3,704	2,855,554	11,970	187	2 6	1 3 1
4	1,206	2,854,794	33	947	2 7	1 3 1

Table 2: Coverage and Effectiveness: comparison of the efficiency of different settings to generate Czechoslovak patterns in 2020 [11]

Word list	Parameters	Good	Bad	Missed	Size	Patterns
2020	custom2020	99.67%	0.00%	0.33%	40 kB	7,417
2020	correctopt2020	99.99%	0.00%	0.01%	45 kB	8,231
2020	sizeopt2020	99.87%	0.03%	0.13%	32 kB	5,907

Table 3: Generalization: results of 10-fold cross-validation with evaluated parameters

Wordlist	Parameters	Good	Bad	Missed
2020	custom2020	99.85%	0.22%	0.15%
2020	correctopt2020	99.95%	0.15%	0.05%
2020	sizeopt2020	99.58%	0.18%	0.42%

wordlist. These points are typically filtered out during typesetting by setting of both hyphenmin registers to 2. We call the new wordlist dataset model 2023uniqlr1: it comes with slightly changed syllable markup and word deduplication.

The results are provided in tables 4 and 5 on the next page. The change gives better coverage metrics but slightly worse generalizations, probably because of introducing other inconsistencies.

“If I had more time I would have written you a shorter letter.”
– Blaise Pascal

5 Parameter Optimization of Pattern Generation

The quality and effectiveness of generating patterns depend on parameters of patgen for generation. There is not much insight and heuristics on how to set up patgen parameters. The most basic hyperparameter tuning method is setting

Table 4: The effect of consistency: statistics from the generation of Czechoslovak hyphenation patterns with consistent syllable markup added for one character syllables at the beginning of words and \lefthyphenmin and \righthyphenmin set to 1 (patgen generation parameters (correctopt2020))

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	2,675	1,605,899	127,339	61,344	1 3	1 5 1
1	1,505	1,604,012	1,883	63,231	1 3	1 5 1
3	4,289	1,667,204	5,390	39	2 6	1 3 1
4	723	1,666,990	3	253	2 7	1 3 1

Table 5: The effect of consistency on generalization: results of 10-fold cross-validation with evaluated parameters

Parameters	Good	Bad	Missed	Size	Patterns	Precision	$F_{1/7}$
custom2020	99.40%	0.75%	0.60%	29 kB	5,124	0.9925	0.9925
correctopt2020	99.57%	0.83%	0.42%	50 kB	8,384	0.9916	0.9917
sizeopt2020	99.11%	0.72%	0.88%	35 kB	5,955	0.9927	0.9927

Table 6: Parameters found by grid search on wordlist dataset model 2023uniqlr1. Generalization metrics: Good: 99.60%, Bad: 0.86%, Missed: 0.40%, Precision: 0.9914, Recall: 0.9960, F-Score ($\beta = 1/7$): 0.9915, F-Score ($\beta = 1/77$): 0.9914

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	2,216	1,615,261	187,508	51,982	1 3	1 4 1
2	1,726	1,612,057	1,896	55,186	1 3	1 4 1
3	4,198	1,667,198	2,647	45	2 6	1 4 1
4	474	1,667,112	0	131	2 7	1 4 1

a grid search. Grid search is a method to perform hyperparameter optimization, that is, it is a method to find the best combination of hyperparameters. Given the exponential growth of setting combinations, at least hopeful parameter combinations are evaluated.

In tables 6 and 7 we report the best pattern generation parameters found in our limited grid search. By changing the linear factor of the number of bad hyphenation points we achieved our best setup with $F_{1/7}$ -scores above .9916.

Table 7: Parameters found by grid search on wordlist dataset model 2023uniqlr1. Generalization metrics: Good: 99.58%, Bad: 0.86%, Missed: 0.42%, Precision: 0.9915, Recall: 0.9958, F-Score ($\beta = 1/7$): 0.9916, F-Score ($\beta = 1/77$): 0.9915

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	1,889	1,625,346	276,301	41,897	1 3	1 3 1
2	1,872	1,620,864	4,504	46,379	1 3	1 4 1
3	3,886	1,667,204	5,414	39	2 6	1 3 1
4	729	1,666,973	0	270	2 7	1 4 1

“While AI programs try to understand sentences by analyzing word patterns, we try to hyphenate words by analyzing letter patterns.” – Frank Liang [6, page 42]

6 Future Work

The feasibility of universal patterns that comprise information for several languages has been confirmed in [7]. Extending Czechoslovak dataset for other Slavic languages, and generating universal Slavic hyphenation is in progress.

A grid search strategy might be found to minimize the size of the pattern set. The success of reduction of the minimal set cover problem to a dictionary problem solvable with competing patterns would lead to the falling of algorithmic barriers [5]. We are trying to find a monotonous ordering of the set of subsets that minimally covers the original set with methods from [2]. Is $P=NP$?

“But in the endgame of life, I fundamentally believe the key to happiness is letting go of that idea of perfection.” – Debra Messing

7 Conclusion

We have studied the possibilities for improvement of machine learning of competing patterns. We have confirmed the necessity of model development and consistency markup in the input dataset. We have shown that techniques like grid search may improve efficiency even further.

We have used the techniques for the development of Czechoslovak hyphenation patterns. The patterns have been deposited on the LINDAT repository <https://lindat.cz>.

Acknowledgement This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ infrastructure LM2023062. We are indebted to Don Knuth for the questioning that has led us in this research direction. Firstly, questioned the common properties of Czech and Slovak hyphenation during our presentation of [10] at TUG 2019. Secondly, he mentioned the $P=NP$ problem during his talk at the Faculty of Informatics MU the same year [13].

References

1. Franceschini, G., Grossi, R., Munro, J., Pagli, L.: Implicit B-trees: a new data structure for the dictionary problem. *Journal of Computer and System Sciences* **68**(4), 788–807 (2004). <https://doi.org/https://doi.org/10.1016/j.jcss.2003.11.003>, special Issue on FOCS 2002
2. Graham, R.L., Knuth, D.E., Patashnik, O.: *Concrete Mathematics*. Addison-Wesley, Reading, MA, USA (1989)
3. Haralambous, Y.: A Revisited Small Tutorial on Patgen, 28 Years After. In electronic form, available from CTAN as `info/patgen2.tutorial` (Mar 2021), <https://mirrors.nic.cz/tex-archive/info/patgen2-tutorial/patgen2-tutorial.pdf>
4. Knuth, D.E.: *Sorting and Searching, The Art of Computer Programming*, vol. 3. Addison-Wesley, third edn. (1998)
5. Knuth, D.E., Daylight, E.G.: *Algorithmic Barriers Falling: P=NP?* Lonely Scholar, Geel, Belgium (2014)
6. Liang, F.M.: *Word Hyphenation by Computer*. Ph.D. thesis, Dept. of Computer Science, Stanford University (Aug 1983), <https://tug.org/docs/liang/liang-thesis.pdf>
7. Sojka, O., Sojka, P., Máca, J.: A roadmap for universal syllabic segmentation. *TUGboat* **44**(2) (2023), <https://doi.org/10.47397/tb/44-2/tb137sojka-syllabic>
8. Sojka, P.: Competing Patterns for Language Engineering. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *Proceedings of the Third International Workshop on Text, Speech and Dialogue—TSD 2000*. pp. 157–162. LNAI 1902, Springer-Verlag, Brno, Czech Republic (Sep 2000). https://doi.org/10.1007/3-540-45323-7_27
9. Sojka, P.: *From Minds to Pixels and Back (Habilitation Thesis)*. Masaryk University, Brno (Apr 2008)
10. Sojka, P., Sojka, O.: The Unreasonable Effectiveness of Pattern Generation. *TUGboat* **40**(2), 187–193 (2019), <https://tug.org/TUGboat/tb40-2/tb125sojka-patgen.pdf>
11. Sojka, P., Sojka, O.: Towards New Czechoslovak Hyphenation Patterns. *Zpravodaj ČSTUG* **30**(3–4), 118–126 (2020). <https://doi.org/10.5300/2020-3-4/118>, <https://cstug.cz/bulletin/pdf/2020-3-4.pdf#page=16>
12. Sojka, P., Sojka, O.: New Czechoslovak Hyphenation Patterns, Word Lists, and Workflow. *TUGboat* **42**(2) (2021), <https://doi.org/10.47397/tb/42-2/tb131sojka-czech>
13. Szaniszló, T.: Dva bloky otázok a odpovedí od Donalda Knutha na FI MU. (Two questions and answers sessions by Donald Knuth at FI MU). *Zpravodaj ČSTUG* **30**(1–2), 64–97 (2020). <https://doi.org/10.5300/2020-1-2/64>

Verb-Object Collocations in the Russian Collocations Database: Linguistic and Statistical Properties

Maria Khokhlova 

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

Abstract. Russian Collocations Database comprises collocations extracted from nine dictionaries. The examples provide additional statistical information based on text corpora. The paper deals with those new characteristics that have been added to the database, and how the verb-object collocations that are represented in it intersect with corpus data. The database offers two kinds of interfaces that imply a simple or an advanced search. The former is aimed at language users while the latter can be used by linguists and show a wide range of quantitative characteristics. The paper also presents results of correlation analysis made between collocation lists extracted from dictionaries and corpora. Verb-object collocations from the top of the list of any association measure used in the database proved to be described in several dictionaries compared to the bottom of the list. Verbs tend to be more productive than nouns and produce more examples.

Keywords: Collocation, database, Russian language, dictionaries, corpora, gold standard

1 Introduction

The Russian Collocations Database is a collection of collocations extracted from Russian explanatory and specialized dictionaries, supplemented with statistical information based on text corpora [1]. Since creating a resource is always a process of trial and error, this paper will focus on what features have been added to the database since the launch of the project and how it has been changed and improved. As an example, we will consider verb-object collocations registered in the database, and they will also be compared with corpus data. Collocations were extracted from a number of acknowledged dictionaries. However, the question arises: how do these dictionary collocations correlate with corpora (first of all, with large web ones).

The paper has the following structure. The Introduction explains the motivation of the paper. Section 2 gives an overview of the enhanced database and its interfaces. The next two sections discuss statistical properties of collocations and analyze their structure, paying attention to quantitative properties. The last section concludes the paper and proposes plans for future work.

2 Database

2.1 Overview

Since the Russian Collocations database was launched, it has been enriched with further examples from other Russian dictionaries. The initial volume was equal to 20,000 units that were described in five dictionaries [2]. At the moment, the database has doubled its size and has about 40,000 collocations, which were extracted from nine lexicographic resources.

Below we will discuss the example of verb-object collocations. The database comprises 20,145 entries of such a type that were obtained from the following six dictionaries (Table 1) [3,4,5,6,7,8].

Table 1: The number of the extracted data per dictionaries.

Borisova, 1995	Mel'čuk et al., 1984	MAS	Reginina et al., 1980	Biriuk et al., 2008	Deribas, 1983
3,908	1,797	3,308	1,832	5,951	8,607

It can be seen that the dictionary of verb-noun collocations by Deribas [5] is the most numerous source in its examples. The maximum number of verb-object collocations is 5 (that is, no collocation occurs in 6 dictionaries), while the maximum value in the case of adj-noun collocations is 6 [9]. The introduced dictionary index indicates the number of dictionaries in which collocations are presented (Table 2). Thus, in case of verb-object collocations, the index ranges from 1 up to 5.

In all 5 dictionaries, we find the following 8 examples: *oderzhat' pobedu* 'to win', *pol'zovat'sya doveriyem* 'to enjoy confidence', *prinyat' mery* 'take measures', *vesti bor'bu* 'to struggle', *ispytyvat' chuvstvo* 'to feel', *nesti otvetstvennost'* 'to be responsible', *pol'zovat'sya uvazheniyem* 'to be held in respect' and *brat' primer* 'to follow the example'. 181 collocations have the dictionary index equal to 4. Almost all of them are given in the dictionaries of collocations by Borisova [4] and of Russian verbal collocability compiled by Biriuk et al [3]. Three dictionaries present 759 common examples, while two resources produce 3,165 phrases. 80% of the total number of verb-object collocations (16,032) is described only in one dictionary.

The items are also represented by longer collocations with objects represented by prepositional or noun phrases. For example, *ispol'zovat' administrativnyy resurs* 'to use administrative resource', *nakopit' opredelennyi opyt* 'to gain certain experience', *podvergnut'sya radiatsionnomu vozdeystviyu* 'to be exposed to radiation', *poluchit' finansovyyu podderzhku* 'to receive financial support', *prinyat' okonchatel'noye resheniye* 'to make a final decision', *slat' serdechnyy privet* 'to send warmest regards'. It is peculiar that all these examples of distance collocations are given in [3].

Table 2: Examples of collocations with different dictionary indices.

Dictionary Collocation	Borisova, 1995	Mel'čuk et al., 1984	MAS	Reginina et al., 1980	Biriuk et al., 2008	Deribas, 1983	dictionary index
<i>oderzhat' pobedu</i> 'to win'	1 ¹	1	1	0	1	1	5
<i>nesti otvetstvennost'</i> 'to be responsible'	1	1	0	1	1	1	5
<i>brat' primer</i> 'to follow the example'	1	0	1	1	1	1	5
<i>stavit' zadachu</i> 'to put the problem'	1	0	0	1	1	1	4
<i>proizvesti vpechatleniye</i> 'to make an impression'	1	1	0	0	1	1	4
<i>otdavat' dan'</i> 'to pay tribute'	0	0	1	1	1	1	4
<i>propvat' blokadu</i> 'to run the blockade'	0	1	0	0	1	1	3
<i>vyderzhat' ekzamen</i> 'to pass the exam'	1	0	0	1	1	0	3
<i>otvodit' vzglyad</i> 'to look away'	1	0	0	0	1	1	3
<i>dostignut' tseli</i> 'to succeed'	1	1	0	0	0	0	2
<i>zavyazat' besedu</i> 'to make a talk'	1	0	0	0	0	1	2
<i>sgladit' ugly</i> 'to smooth things over'	0	0	1	0	0	1	2
<i>zaklyuchit' dogovor</i> 'to enter into a contract'	0	0	0	0	1	0	1
<i>nosit' otpechatok</i> 'to imprint'	0	0	0	0	0	1	1
<i>razgonyat' tosku</i> 'to dispel gloom'	1	0	0	0	0	0	1

Pairwise comparison between the dictionaries has shown that the following resources have the largest intersection (Table 3): 1) dictionaries [5] and [4]; 2) dictionaries [5] and [3].

Table 3: Pairwise comparison between the dictionaries.

	Mel'čuk et al., 1984	MAS	Reginina et al., 1980	Biriuk et al., 2008	Deribas, 1983
Borisova, 1995	124	65	359	313	502
Mel'čuk et al., 1984		11	14	88	222
MAS			9	72	345
Reginina et al., 1980				74	261
Biriuk et al., 2008					706

2.2 Interfaces

Since the database can be in demand by different groups of users, there are two kinds of interfaces, namely, a linguistic search and a statistical one. The first type of interface makes it possible to view the collocations for either a node or collocate. The results contain a list of collocations, in which the following linguistic information is presented:

- definition of lemmata from the Wiktionary;
- type of syntactic structure (i.e., adj-noun, verb-noun, etc.);
- a link to an example of usage in the Russian National Corpus [10];
- presence/absence of a collocation in the SynTagRus [11] and Taiga [12] corpora;
- intersection with other collocations.

The results involve a dictionary index as well. The larger it is, the greater the probability of using a collocation is. We introduced a graphical interpretation of dictionary indices to indicate that a collocation is typical. Figure 1 shows a bar plot for the results for the verb *obratit'* 'to turn'. One can note that the most common examples shown in the dictionaries are *obratit' vnimaniye* 'to draw attention, to give attention' (in three dictionaries), *obratit' vzor* 'to look' and *obratit' v begstvo* 'to put to flight' (both collocations are described in two dictionaries).

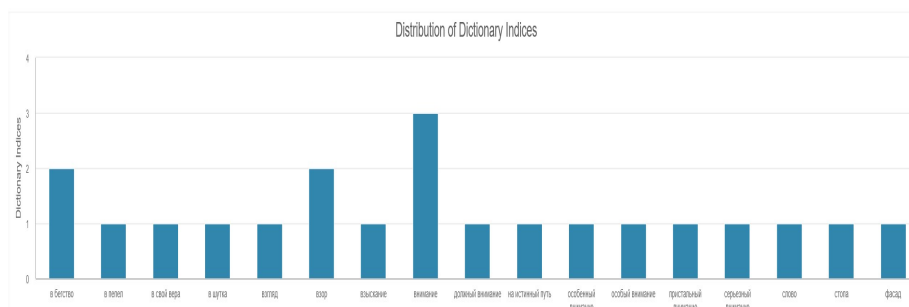


Fig. 1: Distribution of dictionary indices for the verb *obratit'* 'to turn'

Figure 2 shows visualisation used in the database to present the node and its collocates. The examples found in several dictionaries are marked with dark arrows between the verb and its collocates.

A statistical search offers a more specialized way to present results aimed at advanced users. Each entry is supplied with the following statistical information:

- presence of a particular collocation into dictionaries (9 dictionaries in total);
- a dictionary index;

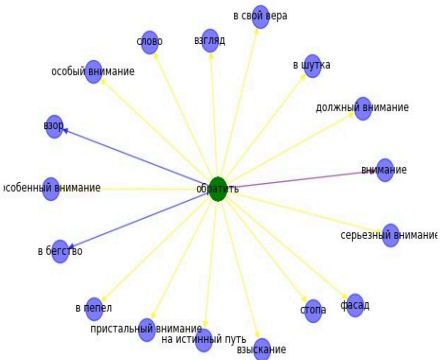


Fig. 2: Distribution of dictionary indices for the verb *obratit'* ‘to turn’

- relative frequency (ipm) based on Russian National corpus and Araneum Russicum Maximum corpus [13];
- values of association measures based on Araneum Russicum Maximum (MI, MI3, log-likelihood, logDice, t-score).

Figures 3 and 4 show examples of the collocations for the verb *igrat'* ‘to play’. Here we find the following examples: *igrat' rol'* ‘to play a role’ (in four dictionaries), *igrat' slovami* ‘to play on words’ (in two dictionaries), *igrat' spektakl'* ‘to play a play’ (in two dictionaries), *igrat' svad'bu* ‘to celebrate a wedding’ (in two dictionaries).

Collocate	Borisova	Melchuk	Kustova	Ubin	MAS	Reginina	BTS	Biryuk	Deribas
роль	1	0	0	0	1	1	0	1	0
свадьба	1	0	0	0	1	0	0	0	0
слово	0	0	0	0	1	0	0	1	0
спектакль	1	0	0	0	0	0	0	1	0
огромный роль	0	0	0	0	0	0	0	1	0
основной роль	0	0	0	0	0	0	0	1	0
особый роль	0	0	0	0	0	0	0	1	0
первый скрипка	0	0	0	0	1	0	0	0	0
песня	0	0	0	0	1	0	0	0	0
решающий роль	0	0	0	0	0	0	0	1	0

Fig. 3: The first part of the output of the statistical search for the verb *igrat'* ‘to play’.

0 and 1 indicate if the collocation is present or absent in the dictionary. The table is the same for all types of collocations and hence shows many zeros if the

Dictionary Index ▼	RNC ipm	Araneum ipm	t-score	MI	MI3	log-likelihood	logDice
4	0	4812.04	401.20999	7.71938	42.33968	1429203.84277	9.4095
2	0	44.36	33.31437	2.84422	23.94187	3329.63306	3.36302
2	0	0.12	-6.2832	-2.05019	1.94981	13.76611	-4.8293
2	0	0.12	1.88748	4.15181	8.15181	15.48438	-4.79447
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Fig. 4: The second part of the output of the statistical search for the verb *igrat* ‘to play’.

example is not given in a dictionary. Six collocations given in Figure 2 were not found in the following dictionaries [5,7,14] and [15].

The second part of the statistical interface shows quantitative data for collocations (see Figure 4 for possible collocates with the verb *igrat* ‘to play’).

3 Statistical Properties and Representation

Statistical validation of the gold standard is an essential step in a database design, as the dictionaries are the product of introspection. Association measures and dictionary indices were used to determine the typical character of word combinations. As statistical indicators, we used highly widespread association measures based on the Araneum Russicum Maximum corpus, namely, MI, MI3, log-likelihood, logDice and t-score. These measures belong to different classes and therefore can produce different results. By interpreting quantitative data, an advanced user can get more thoughtful data. Below we will show which verb-noun collocations are the most frequent when using statistics.

T-score and MI tend to show opposite results [16], and our study confirms this statement. T-score ranges the following collocations as the most typical ones: *igrat’ rol’* ‘to play a role’, *pol’zovat’sya populyarnost’yu* ‘to be popular, in favour’, *udelyat’ vnimaniye* ‘to give attention’, *prinyat’ resheniye* ‘to make a decision’, *oderzhat’ pobedu* ‘to win’. Top-50 includes collocations with the nouns *vnimaniye* ‘attention’ (6²), *populyarnost’* ‘popularity’ (2) and *rol’* ‘role’ (3). The most frequent collocations, selected according to the values of this measure, are recorded on average in two dictionaries. One can suggest a correlation between dictionary and corpus data. In other words, t-score can be used to select data for compiling a dictionary and will show the most frequently occurring examples. MI made it possible to find collocations that occur on average in one dictionary. They are represented by the following examples: *tochit’ lyasy* ‘to chat’, *zamorit’ chervyachka* ‘to have a snack’, *zamolvit’ slovechko* ‘to put in a word (for)’, *porot’*

² Henceforth, the number of collocations is shown in parentheses.

goryachku ‘to be in a hurry’, *smorozit’ glupost’* ‘to say stupid things’. Both a node and a collocate in each example have a low frequency in the corpus and hence collocations are closer, rather, to idioms or phraseological units due to their non-compositionality. As for the three remaining measures (MI3, log-likelihood, logDice), they produce similar results. On average, collocations occur in one or two dictionaries. The most frequent ones are as follows: *oblizyvat’ pal’chiki* ‘about smth delicious’, *igrat’ rol’* ‘to play a role’, *pol’zovat’sya populyarnost’yu* ‘to be popular, to be in favour’, *privlech vnimaniye* ‘to attract attention’, *vyzvat’ interes* ‘to provoke interest’.

One can notice the following trend: top results for the measure are phrases recorded in different dictionaries. These are more frequent collocations both in terms of statistics and in terms of their reproducibility in speech.

For a visual evaluation of collocations, we used bar plots. Figure 5 shows the values of statistical measures (logDice, MI, MI3, t-score) obtained by collocations with the noun *glupost’* ‘stupidity, stupid things’. The highest value for MI3 (the second bar in each group) is equal to 21.94 and corresponds to the collocation *nadelat’ glupost’* ‘to have stupidity’.

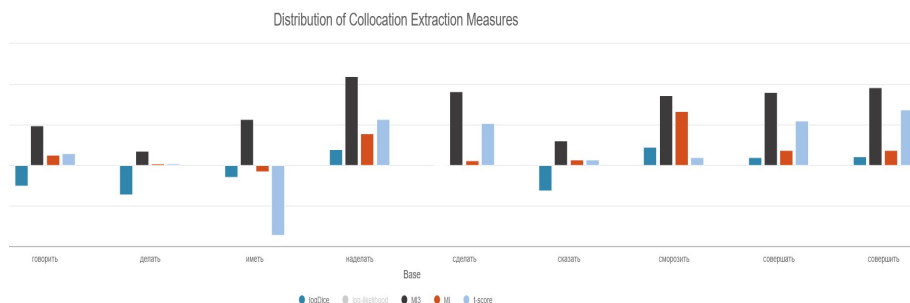


Fig. 5: Distribution of association measures for the noun *glupost’* ‘stupidity, stupid things’

Based on the bar height of the corresponding measure, one can judge how typical a collocation is. For example, *nadelat’ gluposti* ‘to do stupid things’ or *sovershat’/sovershit’ gluposti* ‘to do stupid things’ are frequent, while *smorozit’ glupost’* ‘to say stupid things’ (with the highest value for the MI measure) is almost a phraseological unit. In contrast, negative values for bar plots indicate that collocation is not as common in corpora, despite being registered in dictionaries. Here we can also name *imet’ glupost’* ‘to have stupidity’ (t-score is equal to -17.16) or *delat’ glupost’* ‘to do stupid things’ (logDice is equal to -7.21).

4 Analysis of Dictionary Collocations

Verb-object collocations involves 2,722 verbs, 1,028 (about 38%) among them produce only one collocation. Opposed to adj-noun collocations, verbs are

highly productive. Five verbs form more than 200 collocations : *byt'* 'to be' (223), *davat'* 'to give' (216), *dat'* 'to give' (270), *poluchit'* 'to receive' (271), *sdelat'* 'to do, to make' (223). Other productive verbs can be exemplified by the following ones: *imet'* 'to have' (195), *poluchat'* 'to receive' (192), *delat'* 'to do, to make' (187), *prinyat'* 'to receive' (181), *brat'* 'to take' (177), *vzyat'* 'to take' (137), *provesti* 'to conduct, to lead' (134), *provodit'* 'to conduct, to lead' (132), *vesti* 'to conduct, to lead' (131), *vyzvat'* 'to cause' (124), *vyzyvat'* 'to cause' (123), *videt'* 'to see' (121), *proyavlyat'* 'to display, to show' (120), *prinimat'* 'to receive' (119) and *proyavit'* 'to display, to show' (117).

The list of collocates includes 5,665 nouns in total, of which 1,030 (i.e., about 18%) are unique and form only one collocation. The rest of nouns suggest various collocations, exceeding several dozens. The most productive lexemes are, for example, *zhizn'* 'life' (115), *sila* 'force, power' (103), *delo* 'case, matter' (103), *slovo* 'word' (97), *rabota* 'job, work' (85), *vremya* 'time' (83), *vzglyad* 'glance, opinion' (83), *vopros* 'question' (75), *vozmozhnost'* 'opportunity, possibility' (71), *pravo* 'right' (68), *vnimaniye* 'attention' (64), *interes* 'interest' (63), *polozheniye* 'position' (62), *otnosheniye* 'attitude, relation' (58), *chuvstvo* 'feeling' (56), *nadezhda* 'hope' (56), *mysl'* 'idea, thought' (55), *glaz* (53), *initsiativa* 'initiative' (52) and *vlast'* 'authority' (51).

It should be noted that, unlike nouns, verbs show more significant variability in producing collocations. On average, there are 7.4 collocations per verb, while there are 3.6 collocations per noun.

5 Conclusion and Future Work

In the paper, we discussed the features of the Russian collocations database and analyzed the examples of verb-object collocations. We traced the possible correlation between dictionaries and statistical coefficients. It can be noted that collocations from the top of the list of any measure are more stable and are described in several dictionaries. In verb-object collocations, verbs tend to be more productive than nouns and produce more examples.

We have shown some technical details concerning the database. Visualisation helps users to understand the usage of collocations in speech: how frequent and typical they are. However, it is necessary to enhance the results. Output collocations are shown for lemmas, while it is better to display them as token collocations. There are many zeros in the tables indicating that many phrases are recorded only in one dictionary. Such a table view can be questioned if it is appropriate and user-friendly and might be changed in future.

Acknowledgements. The presented research was supported by the Russian Science Foundation, project No. 22-18-00189 "Structure and functionality of stable multiword units in Russian everyday speech".

References

1. Russian Collocations Database, <http://collocations.spbu.ru>. Last accessed 5 Nov 2023
2. Khokhlova, M.: Collocations in Russian Lexicography and Russian Collocations Database. In: Proceedings of The 12th Language Resources and Evaluation Conference. Marseille, France, pp. 3198–3206. European Language Resources Association (2020).
3. Biriuk, O.L., Gusev, V.Yu., Kalinina, E.Yu.: Dictionary of Russian Abstract Nouns' Verbal Collocability. A Dictionary based on the Russian National Corpus [Slovar' Glagol'noj Sochetamosti Nepredmetnykh Imen Russkogo Yazyka. Slovar' na osnove Natsional'nogo Korpusa Russkogo Yazyka] (2008)
4. Borisova, E.G.: A Word in a Text. A Dictionary of Russian Collocations with English-Russian Dictionary of Keywords [Slovo v tekste. Slovar' kollokatsiy (ustoychivyykh sochetaniy) russkogo yazyka s anglo-russkim slovarem klyuchevyykh slov]. Filologiya, Moscow (1995)
5. Deribas, V.M.: Verb-Noun Collocations in Russian. [Ustoychivyye glagol'no-imennyye slovosochetaniya russkogo yazyka]. Russkij yazyk, Moscow (1983)
6. Dictionary of the Russian Language in 4 volumes [Slovar' russkogo yazyka v 4 tomakh] (MAS) (1999), Yevgen'yeva A. P. (ed.-in-chief). Vol. 1–4, 4th edition, revised and supplemented. Russkij yazyk, Moscow.
7. Mel'čuk, I., Zholkovsky, A.: Explanatory Combinatorial Dictionary of Modern Russian [Tolkovo-kombinatornyy slovar russkogo yazyka]. Vienna (1984)
8. Reginina, K.V., Tjurina, G.P., Shirokova, L.I.: Set Expressions of the Russian Language. A Reference Book for Foreign Students [Ustoychivyye slovosochetaniya russkogo yazyka: Uchebnoye posobiye dlya studentov-inostrantsev], Shirokova, L.I. (ed.). Moscow (1980)
9. Khokhlova, M.: Attributive collocations in the Russian gold standard and their representation in dictionaries and corpora [Attributivnyye kollokatsii v zolotom standarte sochetajemosti russkogo yazyka i ih predstavleniye v slovarjah i korpusah tekstov]. Questions of Lexicography (21), 33–68 (2021)
10. Russian National Corpus, <http://ruscorpora.ru>. Last accessed 5 Nov 2023
11. SynTagRus, <https://ruscorpora.ru/new/search-syntax.html>. Last accessed 5 Nov 2023
12. Taiga, https://tatianashavrina.github.io/taiga_site. Last accessed 5 Nov 2023
13. Benko, V.: Aranea Yet Another Family of (Comparable) Web Corpora. In: Text, Speech and Dialogue. 17th International Conference, TSD 2014, Brno, Czech Republic, September 8–12, 2014. Proceedings. LNCS, vol. 8655, pp. 257–264. Springer, Heidelberg (2014)
14. Kustova, G.I.: Dictionary of Russian Idiomatic Expressions [Slovar' russkoyj idiomatiki. Sochetaniya slov so znacheniyem vysokoy stepeni] (2008).
15. Big Russian explanatory dictionary [Bol'shoy tolkovyy slovar' russkogo yazyka] (BTS), Kuznetsov, S.A. (ed.). Norint, St. Petersburg (1998)
16. Gablasova, D., Brezina, V., McEnery, T.: Collocations in Corpus-Based Language Learning Research: Identifying, Comparing, and Interpreting the Evidence. Language Learning, 67(S1), 155–179 (2017)

Part IV

Semantics and Language Modelling

Towards Using Speech Melody to Guide Large Language Models

David Porteš

Faculty of Informatics,
Masaryk University,
Botanická 68a, 602 00 Brno, Czech Republic
xportes@fi.muni.cz

Abstract. We propose an extension to the standard text prompt-based way of interacting with Large Language Models (LLMs), which allows the user to specify a melody (F0 curve) that the generated text should fit. This novel task, which we call Melodic Alignment, involves guiding the text generation process of an arbitrary LLM by the F0 curve, provided by the user as an additional input in the form of an audio recording of either regular speech or a hummed melody. We also propose an ASR-based benchmark to evaluate the performance of various approaches to solving the Melodic Alignment task, along with a discussion on its potential use cases.

Then, as a first-shot model architecture, we propose Melodic Aligner (Meligner). Meligner uses a separate seq2seq model trained on melody / text parallel data to assign a score to each candidate token during the LLM's decoding phase. The 'F0 fit score' each token receives is then merged with the token's probability assigned by the LLM. The degree to which the generated text is influenced by the rescoring can be tweaked by choosing the appropriate merging strategy.

Samples and other supplementary materials will be added to the following link: https://gitlab.fi.muni.cz/nlp/melodic_alignment

Keywords: LLM, large language model, F0, speech melody, speech translation

1 Introduction

Large Language Models (LLMs) have seen a great surge in popularity in recent time, and have been adopted by the general public as useful assistants for a broad variety of tasks. However, for some applications, the text-only mode of interaction with LLMs can be limiting, particularly in cases when the generated text should sound a certain way when read out loud. In this paper, we propose an extension to the standard text prompt framework, which allows the user to specify an audio recording of the melody that the generated text should fit. This recording can either contain actual speech, from which the melody is extracted, or can contain a hummed melody, depending on the use case (discussed in

section 4). We call this novel task Melodic Alignment and define an ASR-based metric to be used as a performance benchmark.

We outline multiple use cases where the Melodic Alignment framework could be applied, and, as a first-shot approach, we propose Melodic Aligner (Meligner); an F0 alignment scheme using a separate ‘rescoring’ seq2seq model trained on parallel F0 curve / text data. Meligner extracts the F0 curve from the input recording, converts it into embedding vectors, and subsequently feeds it into the encoder of the rescoring model. During the decoding phase of the LLM, Meligner calculates each candidate token’s likelihood that it would have been generated by the rescoring model. This likelihood is then combined with the token’s original probability assigned by the LLM, forming a final score based on which the decoding continues. If beam search is used, this happens at every step and in each beam.

2 Related work

There is, to the extent of our knowledge, no system capable of guiding the text generation process of LLMs by speech melody, which is why we are proposing it in this paper. However, there are certain similarities between Melodic Alignment and established NLP tasks, particularly automatic song lyrics translation and automatic speech recognition (ASR).

2.1 Automatic speech recognition (ASR)

Automatic speech recognition can be considered an extreme case of the Melodic Alignment task. If we consider the case when the user specifies only the audio recording, with no textual prompt, the LLM will generate text guided only by the F0 curve extracted from the user-provided recording. This case can be thought of as a constrained version of the ASR task, where only the F0 curve of the audio is used as input. In fact, in our proposed evaluation method, we make use of precisely this task, as described in section 3.2.

2.2 Automatic song lyrics translation

Probably the closest established task to Melodic Alignment is the task of translating song lyrics into tonal languages. In tonal languages, such as Mandarin, multiple words are pronounced in the same way, and speakers differentiate between them based on their pitch. Therefore, the words in the translated lyrics have to match the melody of the song to avoid misunderstandings [2].

There are multiple different approaches to automatic song lyrics generation, including into tonal languages, such as [7], [5] or [6]. Probably most similar to Meligner is the pitch alignment-based rescoring used in GagaST [2]. GagaST uses rescoring during the decoding process to guide the generation of song lyrics conditioned on the song melody. It calculates a pitch alignment score

based on the relationship between the song melody and a pre-determined word tone (one of the 4 tones of Mandarin).

Meligner can be seen as a generalization of the pitch alignment score to arbitrary F0 shapes, where instead of four predetermined tones for each word, the tone of each word is determined dynamically by the rescoring model, taking into account previous context as well.

3 Task definition & evaluation

3.1 Task definition

The goal is to rescore the outputs of an LLM during its decoding stage, in order to make the resulting text fit a user-specified melody (F0 curve). The F0 curve can either be extracted from a user-provided audio recording, or provided by the user as a series of frequency values in Hz at a defined sampling rate.

3.2 Task evaluation

As an evaluative benchmark, we propose a scheme utilizing real-world speech / transcription pairs as reference. It can be thought of as a constrained version of the Automatic Speech Recognition (ASR) task, in which only the F0 curve is used from the input speech audio. The evaluative scheme calculates a single benchmarking metric, the average rank-shift score. The average rank-shift score is calculated in the following way.

Given a speech / transcription pair, we provide the speech recording as the audio input and let the LLM generate the first token without any textual prompt. At each decoding step, we define the 'correct' token as the token present in the transcription at the position corresponding to the current decoding step. We then observe whether the rank of the 'correct' token moves up or down after rescoring takes place. A well-performing rescoring scheme should naturally see the words from the transcription climbing up in rank after being rescored. Before moving to the next decoding step, we fix the correct token at the current decoding position to be used as context for the next decoding step.

We define rank-shift as the difference between the 'correct' token's original rank and its rank after rescoring. At the end of the decoding process, the rank-shifts from each step are averaged, and a single average rank-shift value is reported.

4 Use cases

The fact that the Melodic Alignment framework preserves the standard textual prompt of the LLM grants it broad applicability. Any prompt that could benefit from the ability to specify a desired melody the generated text should fit can be considered a use case of Melodic Alignment. In this section, we propose just a few of the possible use cases.

Prompt:

Translate the sentence: "sentence to be translated" into english.
(optionally with preceding context)

Audio input:



Fig. 1: Example prompt and audio input for the emotional speech translation task.

Emotional speech translation (dubbing) Currently, there are two main approaches to speech translation. There is the traditional cascade approach, where the speech is first converted into text, translated within a text-to-text framework, and then resynthesized back into audio. By using the intermediate textual representation, the model has access to the vast text-to-text translation data, and can therefore deliver high-quality translations, however, at the cost of losing the prosodic information of the original utterance [1].

The other approach, end-to-end speech translation, where parallel audio recordings are used to translate directly from source speech to target speech, is able to preserve the original prosodic content, however suffers from data scarcity, since parallel speech-to-speech data are much less numerous than their text-to-text counterpart [1].

By casting the translation problem into the Melodic Alignment framework, it is possible to obtain the best of both worlds. One could use the cascading approach, thus exploiting the vast amount of text-to-text data, while at the same time have the translation process be driven by the F0 of the original speech, thus preserving the prosodic content of the original.

Prompt:

"Continue the text: '*previous context*'"

Audio input:



Fig. 2: Example prompt and audio input for the ASR task.

Automatic Speech Recognition (ASR) in highly noisy settings We hypothesize that when conducting Automatic Speech Recognition under conditions where phonemes are unintelligible, i.e. muffled speech, our approach could lead to generating text that, while not exactly accurate, preserves the intent of the speaker. Whether our approach generates useful transcriptions is up to experimentation.

Copywriting aid As a slightly exotic use case, being able to guide text generation by melody could allow authors to take existing dialogues and write alternative ones with identical emotional content. This could allow one to take, for example, a dialogue about topic A and turn it into a dialogue about topic B while keeping the overall style and emotional impression of the original.

Additionally, when writing, for example, advertisement slogans, Melodic Alignment could make it possible to automatically generate a fitting text to a custom, pre-crafted melody. This could allow copywriters to focus their efforts on crafting the ideal, catchy melody for their slogans, leaving it to the Melodic Alignment model to fill in the words after the fact.

Prompt:

Generate a catchy slogan for a car company.

Audio input:

The pre-crafted melody of the slogan



Fig. 3: Example prompt and audio input for the copywriting aid task.

Automatic song translation As mentioned in section 2.2, Melodic Alignment shares many similarities with the task of translating songs into tonal languages. Naturally, a well-performing Melodic Alignment model could be used to solve this task, however, this use case is left for future work, as it will probably require introducing information on the rhythm and other musical features of the original song.

5 Proposed Method: Meligner

As a first-shot architecture, we propose the Meligner scheme. Meligner employs a separate rescoring model; a seq2seq model trained on F0 curve / transcription pairs. The textual prompt is processed in the standard way by the LLM, while

Prompt:

"Translate the following lyrics into english: *song lyrics*."

Audio input:



Fig. 4: Example prompt and audio input for the automatic song translation task.

the audio input is first preprocessed, and then fed into the rescoring model's encoder. The rescoring model is then used during the decoding stage of the LLM, where it assigns each candidate token a score representing the 'goodness of fit' between the token and the F0 curve. In the following, we begin by a description of the F0 preprocessing stage, then we discuss the rescoring model and, finally, the rescoring process itself. For an overview of the entire scheme, please refer to figure 5.

5.1 F0 extraction & preprocessing

We extract the F0 curve from the input recording using the YAAPT algorithm [3]. Then, since state-of-the-art seq2seq models expect their input to be in the form of embedding vectors, we encode the F0 curve obtained from YAAPT into a series of vectors by the use of a vector quantized variational autoencoder (VQ-VAE).

F0 extraction The YAAPT algorithm outputs a series of integer values from 0 to 400, which represent the F0 values in Hz, sampled at a user-defined sampling rate from the audio recording.

$$F0_curve = m_0, m_1, m_2, \dots, m_{N-1} \in [0, 400]$$

where $N = \text{audio length} / \text{sampling rate}$ The sampling rate is a hyperparameter whose value is up to experimentation.

F0 embedding To convert the series of integers into vector embeddings, we make use of a vector quantized variational autoencoder (VQ-VAE), which is an approach adapted from the paper [4]. The VQ-VAE framework consists of a convolutional encoder, a bottleneck layer, and a decoder. It is trained using the standard autoencoder objective, i.e. it is trained to reconstruct its own input, under the limited resources constraint imposed by the bottleneck.

To obtain a vector representation of the F0 curve, we feed it into the encoder of the VQ-VAE and extract the latent vectors.

$$EF_0(m) = (h_1, \dots, h_{L'}), h_i \in \mathbb{R}^{128}$$

Additionally, the latent vectors are each coerced into one of N codebook vectors, with the codebook size N being a tunable hyperparameter. while, in the original paper, only the indices into the codebook are used for further processing, we use the actual codebook vectors.

5.2 Rescoring model

The F0 embedding vectors obtained from the variational autoencoder are fed into the encoder of the rescoring model. The rescoring model is a seq2seq model trained on F0 embedding / text pairs, which serves as a judge of F0 curve / token fit. To teach the model the correspondence between F0 and tokens, we cast the problem into a translation task. We consider the melody embeddings to be a sort of language, and the task of finding words fitting to a given melody is transformed into the task of translating from this ‘melody embedding language’ to English (or any other language). While any seq2seq model can be used, we use the standard transformer architecture conforming to [8].

5.3 LLM outputs rescoring

The rescoring of LLM outputs during decoding is conducted as follows. At each step of the beam search, and in each of the beams, the rescoring model is used to calculate a score for the top N candidate tokens. This ‘F0 fit score’ is obtained as the likelihood that the token would be generated by the seq2seq model, given the input melody and the preceding text so far generated by the LLM. Each token’s probability assigned by the LLM is then merged with its F0 fit score, and decoding is continued using these new scores.

Tokenization Meligner allows any tokenization to be used, provided that it is shared by both the LLM and the rescoring model. If different tokenization is used for each model, the two tokenization schemes should be compatible in the sense that it should be possible to break down the LLM token into multiple rescoring model tokens. The rescoring model tokens can then be fed sequentially into the rescoring model, with the likelihoods at each step aggregated by multiplying and then taking the n-th root. This is the approach we use in our experiments, as we are using phonemes as our tokens of choice for the rescoring model. We believe that it is at this level of granularity that the relationship between melody and text is the most prominent.

6 Training of models used in Meligner

In order to train the VQ-VAE and the rescoring model, we use a speech / transcription dataset. We split the dataset into two parts, one for training of the VQ-VAE encoder, and the other to train the rescoring model. For an overview of the entire training process, please refer to figure 6.

6.1 F0 encoder (VQ-VAE)

The first split is used to train the F0 encoder. We train the F0 encoder in the same way as [4], i.e. we extract F0 from each recording in the dataset by applying the YAAPT algorithm, and subsequently use it to train the autoencoder, using a reconstruction objective. The transcriptions in the dataset are not used.

6.2 Rescoring model

To train the rescoring model, the other split of the speech / transcription dataset is used. We first convert the dataset into an F0 curve / transcription dataset by extracting the F0 curve from each speech recording using the YAAPT algorithm. Then, we use the F0 encoder we have trained in the previous step to convert the raw F0 curve into a series of vector embeddings. The model is then trained on F0 embeddings/transcription pairs in a standard seq2seq framework.

7 Conclusion

We have proposed Melodic Alignment, a novel task that involves using speech melody, provided as an audio input in addition to the standard textual prompt, to guide the text generation process of Large Language Models (LLMs). We also propose a method for evaluation of performance at the Melodic Alignment task. Since the task is designed to preserve the standard textual prompt of the LLM, its applications are as varied as the number of prompts one can come up with.

Additionally, we have presented a rescoring scheme, Meligner, as a first-shot attempt at solving Melodic Alignment. The scheme is model agnostic and, as such, is compatible with any model architecture. As a next step, we plan to implement the Meligner scheme and evaluate its performance on real-world data, using the proposed evaluation scheme. In case of favorable results, we will apply Meligner to the use cases described in section 4, as well as investigate applying the same approach to other prosodic features, such as stress or rhythm.

References

1. Bentivogli, L., Cettolo, M., Gaido, M., Karakanta, A., Martinelli, A., Negri, M., Turchi, M.: Cascade versus direct speech translation: Do the differences still make a difference? In: Annual Meeting of the Association for Computational Linguistics (2021), <https://api.semanticscholar.org/CorpusID:235293674>

2. Guo, F., Zhang, C., Zhang, Z., He, Q., Zhang, K., Xie, J., Boyd-Graber, J.: Automatic song translation for tonal languages. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) *Findings of the Association for Computational Linguistics: ACL 2022*. pp. 729–743. Association for Computational Linguistics, Dublin, Ireland (May 2022). <https://doi.org/10.18653/v1/2022.findings-acl.60>, <https://aclanthology.org/2022.findings-acl.60>
3. Kasi, K.: Yet another algorithm for pitch tracking (yaapt) by (2002), <https://api.semanticscholar.org/CorpusID:15301096>
4. Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.N., Rahman Mohamed, A., Dupoux, E.: Speech resynthesis from discrete disentangled self-supervised representations. In: *Interspeech (2021)*, <https://api.semanticscholar.org/CorpusID:262491522>
5. Qian, T., Lou, F., Shi, J., Wu, Y., Guo, S., Yin, X., Jin, Q.: UniLG: A unified structure-aware framework for lyrics generation. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 983–1001. Association for Computational Linguistics, Toronto, Canada (Jul 2023). <https://doi.org/10.18653/v1/2023.acl-long.56>, <https://aclanthology.org/2023.acl-long.56>
6. Qian, T., Shi, J., Guo, S., Wu, P., Jin, Q.: Training strategies for automatic song writing: A unified framework perspective. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 4738–4742 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9746818>
7. Tian, Y., Narayan-Chen, A., Oraby, S., Cervone, A., Tao, C., Sigurdsson, G., Zhao, W., Chung, T., Huang, J., Peng, V.: Unsupervised melody-to-lyric generation. In: *ACL 2023 (2023)*, <https://www.amazon.science/publications/unsupervised-melody-to-lyric-generation>
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. p. 6000–6010. NIPS’17, Curran Associates Inc., Red Hook, NY, USA (2017)

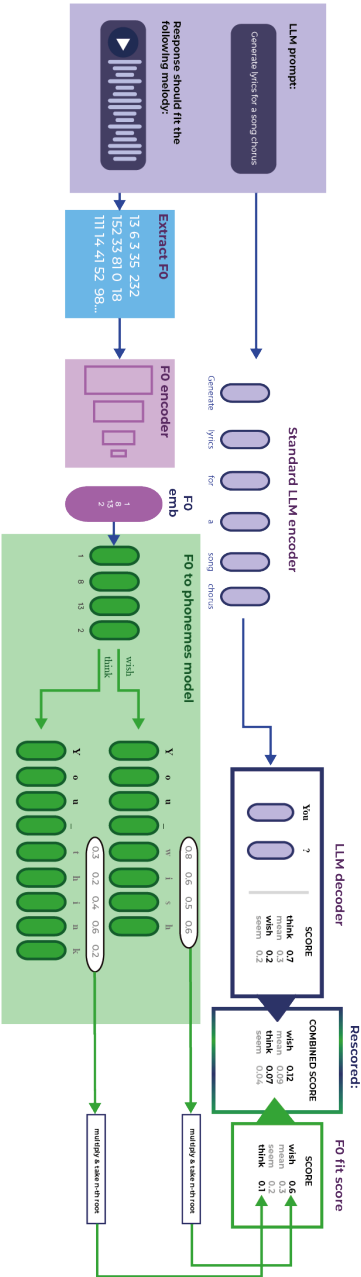


Fig. 5: This figure illustrates the Meligner approach described in section 5.

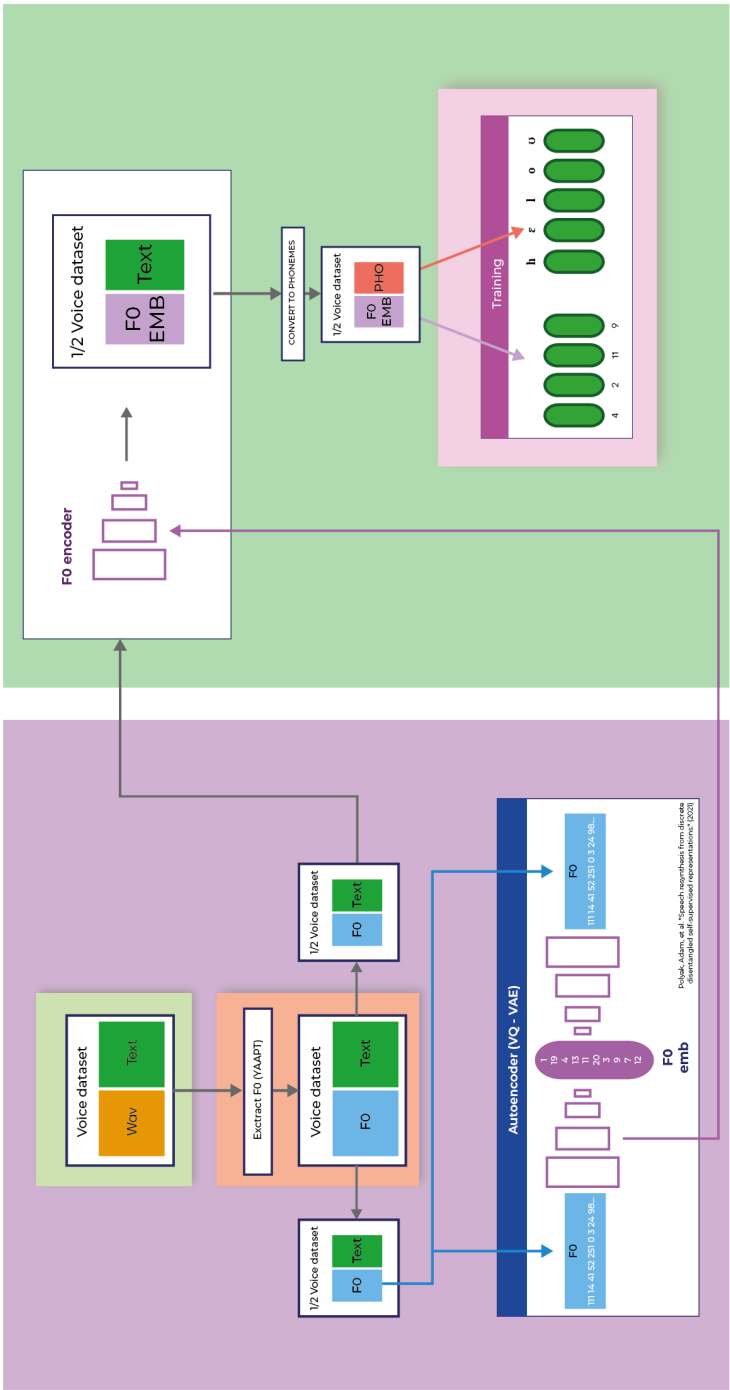


Fig. 6: This figure illustrates the training of the F0 encoder (purple) and the rescore model (green) described in sections 6.1 and 6.2.

Fine-Grained Language Relatedness for Zero-Shot Silesian-English Translation

Edoardo Signoroni

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

Abstract. When parallel corpora are not available to train or fine-tune Machine Translation (MT) systems, one solution is to use data from a related language, and operate in a zero-shot setting. We explore the behaviour and performance of two pre-trained Large Language Models (LLMs) for zero-shot Silesian-English translation, by fine-tuning them on increasingly related languages. Our experiment shows that using data from related languages generally improves the zero-shot translation performance for our language pair, but the optimal fine-grained choice inside the Slavic language family is non-trivial and depends on the model characteristics.

Keywords: machine translation, large language models, English, Silesian, evaluation, zero-shot

Introduction

To date, out of the 7000+¹ languages of the Earth, less than 2%² is covered by the machine translation systems available to the public.

Roughly half of the existing languages do not have any data that can be employed in machine translation [9]. In such cases, one strategy one could employ is to rely on related data and models to operate a zero-shot translation of the resource-scarce language pair. Previous work shows that training or transfer learning among related languages improves the performance for the low-resource pair.

However, most of this work focused on training and then fine-tuning systems from scratch. Language relatedness is also looked at horizontally, usually considering high-level language families. In this paper, we explore relatedness with increasingly fine-grained degree of relatedness with a study inside the Slavic language family, focussing on Silesian-English zero-shot translation.

We fine-tune pretrained multilingual T5 [21] variants, the subword-based mT5 [29] and the byte-level ByT5 [28], enabling for a comparison between the two processing methodologies. We evaluate the output translations with two automated metrics, ChrF++ [20] and COMET [22].

¹ Ethnologue (<https://www.ethnologue.com/>) lists 7168 languages, of which 3072 are endangered.

² As of November 2023, Google Translate supports 133 languages.

We find that using data from related languages generally improves the zero-shot translation performance for our language pair, with the greater improvement between the unrelated language and one from the same high-level language family. The results, however, also show that the behaviour of the models at a finer-grained scale is more complex and depends on the model characteristics.

1 Related Work

1.1 Language Relatedness

Previous studies have investigated language relatedness for transfer learning and MT, with most of the work focussing on training and fine-tuning multilingual models based on the Transformer [27] or Recurrent Neural Networks.

Zoph et al. (2016) [30] show that a French-English parent is better than a German-English one to initialize a Spanish-English model when trying to improve translation quality. Spanish is linguistically closer to French than German.

Dabre et al. (2017) [3] build on the work of Zoph et al. (2016) and expand the experiment in a multilingual setting. They show that transfer learning from an X-Y language pair to a Z-Y language pair has a maximum impact when the second pair is resource scarce and X and Z are in the same or similar language family.

Nguyen and Chiang (2018) [17] improve on the method from Zoph et al. and focuses on exploiting the shared lexicon of related low-resource languages. Their work is made more efficient by Kocmi and Bojar (2018) [11].

Lakew et al. (2019) [14] explore the adaptation of multilingual neural MT models to unseen languages. They find that using language model perplexity as a relatedness proxy to select the most relevant data to the test language improves translation, even in zero-shot situations.

Khatri et al. (2021) [10] focus on Indic languages and show that training a multilingual system on related languages improves the translation performance for their setting.

Edman et al. (2021) [4] applied a novel method for initializing the vocabulary of an unseen low-resource language from a related one, which resulted in an increased translation performance.

1.2 *T5 models

Raffel et al. (2019) [21] describe the “Text-to-Text Transfer Transformer” (T5), a multitask encoder-decoder LLM based on the Transformer architecture. T5 is trained on the “Colossal Clean Crawled Corpus” (C4), a heuristically-cleaned version of the Common Crawl web dump containing about 750GB of English text. T5 uses a unified “text-to-text” format for all text-based NLP problems.

Xue et al. (2021) [29] present mT5, a multilingual variant of T5 trained on a Common Crawl-based dataset covering 101 languages, called mC4. mT5 is

a subword-based model, with a vocabulary of 250k SentencePiece [13] tokens. The authors focus on zero-shot generation with the aim of preventing accidental translation when evaluating generative multilingual LLMs in a zero-shot setting. Both mT5 and its byte-level variant ByT5 have been released in five model sizes: *Small* (300M parameters), *Base* (580M), *Large* (1.2B), *XL* (3.7B), and *XXL* (13B).

Xue et al. (2022) [28] details ByT5, a token-free version of mT5 which works directly on UTF-8 byte sequences, resulting in a vocabulary of 256 possible values, thus reducing the parameters allocated to the vocabulary from 85% to 0.3% for the *Small* model. Therefore, ByT5 can process text in any language, it is more robust to noise, performs better at spelling-sensitive tasks, and does not require complex preprocessing pipelines. It is competitive with subword baselines with 4x less training text, but it has greater training and inference times, due to the increased length of byte sequences.

1.3 Evaluation Metrics

Machine translation is commonly evaluated by comparing the generated text with a reference translation through automated metrics.

ChrF++ [20] is a lexical overlap-based metric includes word bigrams to the character n -gram F-score metric proposed by Popović (2015) [19]. It calculates word and character level F-scores and then averages them together. This metric correlates stronger with human judgements than previous lexical-based metrics, such as BLEU [18] by better matching morphological variants of words.

COMET [22] (Crosslingual Optimized Metric for Evaluation of Translation) is a learned metric originally fine-tuned to estimate a Direct Assessment (DA) score [7] for a given translation by comparing it to source and reference embeddings. It was trained on top of XLM-R-large [2] on a corpus of human judgements of automated translations, both as DA or following the Multidimensional Quality Metric framework [15].

1.4 Parallel Corpora

The MaCoCu project is aimed at building monolingual and parallel corpora for under-resourced European languages by crawling large amounts of textual data from top-level domains of the Internet, and then applying a curation and enrichment pipeline [1]. It covers 17 languages, 8 (Bosnian, Bulgarian, Croatian, Macedonian, Montenegrin, Serbian, Slovene, Ukrainian) are Slavic.

The WikiMatrix [24] project extracted 135 million parallel sentences for 1620 different language pairs using massive multilingual sentence embeddings to automatically extract parallel sentences from the content of Wikipedia articles in 96 languages, including several dialects and low-resource languages. We used the Polish-English section of this corpus.

CzEng 2.0 [12] is an updated version of the CzEng parallel corpus containing 188 million parallel Czech-English sentences spanning multiple sources and domains.

Table 1: Summary of the language selection for the experiment. The last column gives the relatedness degree we assigned to each language, from 0 (completely unrelated) to 4 (closely related). These roughly correspond to the taxonomy of the language with respect to Silesian. Croatian, Serbian, and Ukrainian are on the same level of the taxonomy, but we assigned a higher score to Croatian by virtue of it sharing the same script with Silesian.

Language	ISO Code	Group	Script	Classification
Silesian	szl	West Slavic, Lechitic, Polish-Silesian	Latin	-
Polish	pol	West Slavic, Lechitic, Polish-Silesian	Latin	4
Czech	ces	West Slavic, Lechitic, Czech-Slovak	Latin	3
Croatian	hrv	South Slavic, Western South Slavic	Latin	2
Serbian	srp	South Slavic, Western South Slavic	Cyrillic	1
Ukrainian	ukr	East Slavic, Ukrainian-Rusyn	Cyrillic	1
Maltese	mlt	Afro-Asiatic, Semitic, Arabic, ...	Latin	0

Goyal et al. (2022) [6] release the Flores evaluation benchmark, consisting of 3001 sentences extracted from English Wikipedia translated in 200 languages by professional translators. This enables better assessment of model quality on low-resource languages. We use the Silesian portion as our zero-shot source.

2 Methodology

2.1 Languages, Models, and Metrics Selection

The first step of the experiment consisted in finding a proper dataset that allowed for an as clean as possible comparison. The Flores benchmark dataset features Silesian, a West Slavic language of the Lechitic subgroup, mostly spoken in Upper Silesia, Poland. Joshi et al. (2020) [9] lists Silesian as a low-resource language in terms of availability of data and research.³

To find data for related Slavic languages, we turned to the MaCoCu project, which evaluation⁴ shows it having a significantly better quality than other web-crawled parallel corpora. Following the taxonomy in Glottolog [8], we selected 6 Slavic languages from the corpus, summarized in Table 1. The furthest removed from Silesian are Croatian, Serbian (South Slavic), and Ukrainian (East Slavic). The latter two, being written in Cyrillic script, do not even share the same writing system of Silesian. As our control language, we chose Maltese, a Semitic language also part of the MaCoCu selection.

Since the MaCoCu corpus does not cover any West Slavic language, we had to look elsewhere for languages closer to Silesian. We decided to use Czech as a West Slavic language not belonging to the Lechitic subgroup. We chose to use

³ However, the OPUS repository [26] lists some Silesian-English parallel data available, with the NLLB [25] one consisting of 1.8 million sentences.

⁴ <https://macocu.eu/static/media/second-report.453a82100b1ec3647012.pdf> (Retrieved on Nov 4, 2023)

Table 2: ChrF++ and COMET scores for each system. The best system is given in **bold** and the worst in *italic*.

Fine-Tuning Language	ChrF++		COMET	
	ByT5	mT5	ByT5	mT5
4_pol_Latn	39.6	29.19	0.56	0.45
3_ces_Latn	34.87	28.09	0.48	<i>0.42</i>
2_hrv_Latn	33.22	28.92	0.47	0.47
1_ukr_Cyrl	34.12	29.09	0.5	0.44
1_srp_Cyrl	33.73	29.36	0.5	0.46
0_mlt_Latn	25.43	24.77	<i>0.4</i>	0.44

the CzEng 2.0 parallel corpus. As the closest language to Silesian, we selected Polish, part of the same Polish-Silesian branch of the Lechitic subgroup. The Polish data is taken from WikiMatrix.

With regard to the pre-trained models, we chose mT5-small and ByT5-small. Their similarity in training and architecture allows for a clearer comparison between subword and character-level models. Both were pretrained on the mC4 multilingual corpus, which contains data for some of the languages in our experiments and other Slavic languages in general.

Studies such as the one by Mathur et al. (2020) [16] argue for the retirement of BLEU in favour of ChrF++. Moreover, Sai B. et al. (2023) [23] finds that ChrF++ performs the best among overlap metrics for a selection of Indic languages.

However, both the aforementioned studies and the results of recent WMT Metrics shared tasks [5] demonstrate that learned neural metrics are the most optimal, as they better correlate with human judgements. Among these, COMET is the current state-of-the-art, and is widely employed in machine translation studies.

2.2 Experimental Setup

We first fine-tune translation models from each related language into English on a random sample of 250k sentence pairs. Using the HuggingFace framework, we train for a maximum of 4000 steps with a learning rate of 1e-4 and batches of 5000 tokens, with early-stopping according to the validation performance on the "dev" split of Flores-200.

To evaluate zero-shot performance, we generate English translations for the Silesian "devtest" section of Flores-200 using the fine-tuned model for each language. We then score the output with ChrF++ and COMET, using the implementations provided by HuggingFace.

3 Results

Figure 1 and Table 2 report both the Chrf++ and COMET scores for the zero-shot Silesian-English translation. From the plots, it is clear that the two models

behave quite differently, with ByT5 models almost always performing better than the mT5 ones.

For ByT5, the trend is similar across the two metrics: as expected, the lowest score is for the system trained on Maltese with 25.43 ChrF++ and 0.4 COMET, while the best performance is achieved by the Polish model with 39.6 ChrF++ and 0.56 COMET. Between the two extremes, however, the trend becomes murkier. The performance for the first two related languages, Serbian and Ukrainian is similar, at around 34 ChrF++ and 0.5 COMET, and considerably better than the unrelated language. However, as we move to Croatian, the scores dip to 30.77 ChrF++ and 0.47 COMET. With Czech, the performance increases again to 34.87 ChrF++ and 0.48 COMET. The scores for Croatian and Czech also highlight that this trend seems to be more marked for COMET scores, with the ChrF++ curve being still almost flat.

The behaviour of mT5 is even more complex. According to ChrF++, the only significative jump in performance is between Maltese at 24.77 points and all the Slavic languages, which scores lie around 28/29 points. Interestingly, the best system is the Serbian one, but just for a meagre 0.17 ChrF++. However, the scores for all the Slavic mT5 systems are so close together that no observation apart from that using a Slavic language instead of an unrelated one leads to better zero-shot performance on Silesian.

As with ByT5, the COMET plot for mT5 systems appears to be more varied. Two main points come up: first, the Maltese system performs on-par or even better than some other systems trained on related Slavic languages. It is just 0.1 COMET away from the Polish system, which sits at 0.45 points, and beats the Czech system by 0.2 COMET. Second, the best performance is obtained with Croatian fine-tuning, at 0.47 COMET.

Table 3: Number of tokens (in billions) and representation (as percentage of the training corpus mC4) for Slavic languages and Maltese in ByT5 and mT5. The languages are given in ISO-639-3 codes. Croatian (hrv) is not in mC4.

Language	Tokens (in Billions)	mC4 %
pol	130	2.15
ces	63	1.72
ukr	41	1.51
srp	4.5	0.72
hrv	0	0
all_slavic	1005.09	15.2
mlt	5.2	0.64

Table 3 gives the amount of pretraining data in the mC4 corpus for the relevant languages in our experiment. The amount of seen data for a given language does not seem to strongly impact the performance on zero-shot translation from Silesian. While it is true that Polish is by far the most represented language of the sample in pretraining, it is also the case that the model fine-tuned on Croatian,

which is not present in the mC4 corpus, does not perform significantly worse than the others.

The scores for the fine-tuned systems when translating from the language of training into English is given in Table 4. The quality of the fine-tuned systems on seen source translation similarly does not appear to affect zero-shot translation. While, according to COMET, the performance is roughly at the same level for all systems, looking at ChrF++ gives another picture. As expected, the Maltese ByT5-system seems unable to overcome the typological distance when translating from Silesian, even despite its greater score. The much worse, at least according to ChrF++, Polish ByT5 system is much better for Silesian, losing just 2 ChrF++ points in the zero-shot scenario. This closeness in performance is most probably due to the high degree of relatedness between Polish and Silesian.

Overall, these results seem to indicate that language relatedness plays a part in the zero-shot translation from Silesian to English. Especially for ByT5, it is clear that fine-tuning the system for a related Slavic language improves the translation. While the closest language, Polish, performs the best for ByT5, the same cannot be said for mT5. Moreover, the impact of relatedness on a more fine-grained scale has to be further clarified, with performance fluctuating among the Slavic languages apart from Polish and with the subword model in particular.

4 Conclusions

In this paper, we described our experiment on the impact of related language fine-tuning of multilingual pretrained models for Silesian-English zero-shot translation. We compared the performance of subword-based mT5 and byte-based ByT5 models fine-tuned on a fine-grained selection of increasing related Slavic languages. Using related language data for fine-tuning seems to be beneficial in most of the cases, and while there seems to be an overall upward trend for byte models, the impact of relatedness at a finer-grained scale is still to be clarified. The representation of the fine-tuning language in the pre-trained model and the performance of the fine-tuned system translating from the seen source does not seem to play a part in our zero-shot scenario.

Table 4: ChrF++ and COMET scores for the fine-tuned systems when translating from the language of training into English.

Language	mlt		srp		ukr		hrv		ces		pol	
Model	ByT5	mT5	ByT5	mT5	ByT5	mT5	ByT5	mT5	ByT5	mT5	ByT5	mT5
ChrF++	57.23	43.54	49.34	43.33	46.74	42.64	47.8	38.74	47.85	42.7	41.6	36.29
COMET	0.67	0.57	0.72	0.69	0.72	0.7	0.73	0.64	0.73	0.7	0.7	0.66

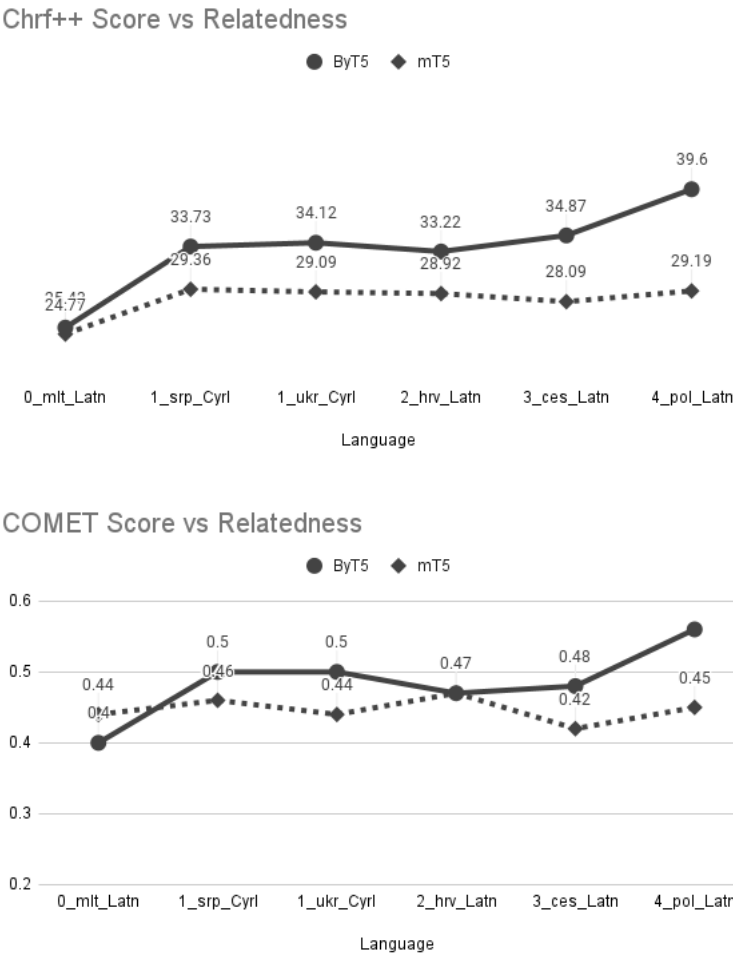


Fig. 1: Plots of ChrF++ and COMET scores for mT5 (dotted line) and ByT5 (full line) models, in order of language relatedness. The left Y-axis reports the scores, while the X-axis gives the fine-tuning language, following the Flores naming conventions. The right Y-axis shows the amount of tokens (in billions) present in the mC4 corpus for each language. The brighter lines represent the score of the fine-tuned system when translating from a seen source.

Limitations and Future work

This work covers just one narrow case of source-side zero-shot translation. The experiment may be expanded to other language pairs and model sizes, since the behaviour of the smaller models may differ from the larger ones.

While we tried to use comparable data from only one source for fine-tuning, for at least two languages, Czech and Polish, this was not completely possible, as they were not covered by the MaCoCu project. This can be an issue, especially with the Polish data, that are exclusively from the same domain as the Flores-200 test set. The Polish systems do not perform consistently better than the others, and thus domain similarity could play a smaller role than anticipated.

Acknowledgments. The work described herein has also been supported by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2023062 LINDAT/CLARIAH-CZ.

References

1. Bañón, M., Esplà-Gomis, M., Forcada, M.L., García-Romero, C., Kuzman, T., Ljubešić, N., van Noord, R., Sempere, L.P., Ramírez-Sánchez, G., Rupnik, P., Suchomel, V., Toral, A., van der Werff, T., Zaragoza, J.: MaCoCu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. In: Proceedings of the 23rd Annual Conference of the European Association for Machine Translation. pp. 303–304. European Association for Machine Translation, Ghent, Belgium (Jun 2022), <https://aclanthology.org/2022.eamt-1.41>
2. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.747>, <https://aclanthology.org/2020.acl-main.747>
3. Dabre, R., Nakagawa, T., Kazawa, H.: An empirical study of language relatedness for transfer learning in neural machine translation. In: Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation. pp. 282–286. The National University (Philippines) (Nov 2017), <https://aclanthology.org/Y17-1038>
4. Edman, L., Üstün, A., Toral, A., van Noord, G.: Unsupervised translation of German–Lower Sorbian: Exploring training and novel transfer methods on a low-resource language. In: Proceedings of the Sixth Conference on Machine Translation. pp. 982–988. Association for Computational Linguistics, Online (Nov 2021), <https://aclanthology.org/2021.wmt-1.104>
5. Freitag, M., Rei, R., Mathur, N., Lo, C.k., Stewart, C., Avramidis, E., Kocmi, T., Foster, G., Lavie, A., Martins, A.F.T.: Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In: Proceedings of the Seventh Conference on Machine Translation (WMT). pp. 46–68. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (Dec 2022), <https://aclanthology.org/2022.wmt-1.2>

6. Goyal, N., Gao, C., Chaudhary, V., Chen, P.J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., Fan, A.: The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics* **10**, 522–538 (2022). https://doi.org/10.1162/tacl_a_00474, <https://aclanthology.org/2022.tacl-1.30>
7. Graham, Y., Baldwin, T., Moffat, A., Zobel, J.: Continuous measurement scales in human evaluation of machine translation. In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. pp. 33–41. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013), <https://aclanthology.org/W13-2305>
8. Hammarström, H., Forkel, R., Haspelmath, M., Bank, S.: *glottolog/glottolog: Glottolog database 4.8* (Jul 2023). <https://doi.org/10.5281/ZENODO.8131084>, <https://zenodo.org/record/8131084>
9. Joshi, P., Santy, S., Budhiraja, A., Bali, K., Choudhury, M.: The state and fate of linguistic diversity and inclusion in the NLP world. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 6282–6293. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.560>, <https://aclanthology.org/2020.acl-main.560>
10. Khatri, J., Saini, N., Bhattacharyya, P.: Language relatedness and lexical closeness can help improve multilingual NMT: IITBombay@MultiIndicNMT WAT2021. In: *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*. pp. 217–223. Association for Computational Linguistics, Online (Aug 2021). <https://doi.org/10.18653/v1/2021.wat-1.26>, <https://aclanthology.org/2021.wat-1.26>
11. Kocmi, T., Bojar, O.: Trivial transfer learning for low-resource neural machine translation. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. pp. 244–252. Association for Computational Linguistics, Brussels, Belgium (Oct 2018). <https://doi.org/10.18653/v1/W18-6325>, <https://aclanthology.org/W18-6325>
12. Kocmi, T., Popel, M., Bojar, O.: Announcing czeng 2.0 parallel corpus with over 2 gigawords (2020)
13. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). <https://doi.org/10.18653/v1/D18-2012>, <https://aclanthology.org/D18-2012>
14. Lakew, S.M., Karakanta, A., Federico, M., Negri, M., Turchi, M.: Adapting multilingual neural machine translation to unseen languages (2019)
15. Lommel, A., Uszkoreit, H., Burchardt, A.: Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica tecnologies de la traducció* (12), 455–463 (Dec 2014). <https://doi.org/10.5565/rev/tradumatica.77>
16. Mathur, N., Baldwin, T., Cohn, T.: Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 4984–4997. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.448>, <https://aclanthology.org/2020.acl-main.448>
17. Nguyen, T.Q., Chiang, D.: Transfer learning across low-resource, related languages for neural machine translation (2017)

18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Isabelle, P., Charniak, E., Lin, D. (eds.) *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002). <https://doi.org/10.3115/1073083.1073135>, <https://aclanthology.org/P02-1040>
19. Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. pp. 392–395. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). <https://doi.org/10.18653/v1/W15-3049>, <https://aclanthology.org/W15-3049>
20. Popović, M.: chrF++: words helping character n-grams. In: *Proceedings of the Second Conference on Machine Translation*. p. 612–618. Association for Computational Linguistics, Copenhagen, Denmark (2017). <https://doi.org/10.18653/v1/W17-4770>, <http://aclweb.org/anthology/W17-4770>
21. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (2019)
22. Rei, R., Stewart, C., Farinha, A.C., Lavie, A.: COMET: A neural framework for MT evaluation. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 2685–2702. Association for Computational Linguistics, Online (Nov 2020). <https://doi.org/10.18653/v1/2020.emnlp-main.213>, <https://aclanthology.org/2020.emnlp-main.213>
23. Sai B, A., Dixit, T., Nagarajan, V., Kunchukuttan, A., Kumar, P., Khapra, M.M., Dabre, R.: IndicMT eval: A dataset to meta-evaluate machine translation metrics for Indian languages. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 14210–14228. Association for Computational Linguistics, Toronto, Canada (Jul 2023). <https://doi.org/10.18653/v1/2023.acl-long.795>, <https://aclanthology.org/2023.acl-long.795>
24. Schwenk, H., Chaudhary, V., Sun, S., Gong, H., Guzmán, F.: WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. pp. 1351–1361. Association for Computational Linguistics, Online (Apr 2021). <https://doi.org/10.18653/v1/2021.eacl-main.115>, <https://aclanthology.org/2021.eacl-main.115>
25. Team, N.: No language left behind: Scaling human-centered machine translation (2022)
26. Tiedemann, J.: Parallel data, tools and interfaces in OPUS. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. pp. 2214–2218. European Language Resources Association (ELRA), Istanbul, Turkey (May 2012), http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
28. Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., Raffel, C.: ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*

- 10, 291–306 (2022). https://doi.org/10.1162/tacl_a_00461, <https://aclanthology.org/2022.tacl-1.17>
29. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021). <https://doi.org/10.18653/v1/2021.naacl-main.41>, <https://aclanthology.org/2021.naacl-main.41>
 30. Zoph, B., Yuret, D., May, J., Knight, K.: Transfer learning for low-resource neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1568–1575. Association for Computational Linguistics, Austin, Texas (Nov 2016). <https://doi.org/10.18653/v1/D16-1163>, <https://aclanthology.org/D16-1163>

Creating an Annotated Health Record Dataset in a Limited-Resource Environment

Kristof Anetta 

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
xanetta@fi.muni.cz

Abstract. This paper demonstrates a workflow for creating a dataset of annotated electronic health records in an environment that is limited in terms of both language resources and expert availability. From preannotation using rule-based methods to the redundancy of multiple annotators per document and the resulting degrees of confidence for each annotation, including the possible avenues of data augmentation in order to be able to train large language models, this paper discusses the practical considerations of how to make the best of the resource-strapped situation shared by so many researchers who analyze health records.

Keywords: Electronic health records, EHR, annotation, named entity recognition, NER, medical concept mining.

1 Introduction

The lack of annotated data is a notorious issue in the field of electronic health record (EHR) analysis. The free-text data of electronic health records is widely considered to be a valuable yet largely untapped resource containing information about both medical science and the populations involved. However, the data exists in a form that cannot be properly understood by common large language models (LLMs) due to their being trained on natural language, not the dense, domain-specific, abbreviated structure of health record text.

While there are powerful LLMs for biomedical text in the English language (such as Gatortron [11] by NVIDIA and the University of Florida, and many others [10]), the situation in small languages such as Czech is dire and not likely to improve in the near future. This is due to the fact that there are no publicly available databases of health records and very few have been made available even to research teams. Adding to that, Czech does not have a large representation in the Unified Medical Language System (UMLS), making even vocabulary-based methods difficult. From the point of view of medical language processing, Czech can be considered a low-resourced language, and just like in so many other languages of similar size, there is no easy way of computationally locating medical concepts in free text - it needs to be annotated using human

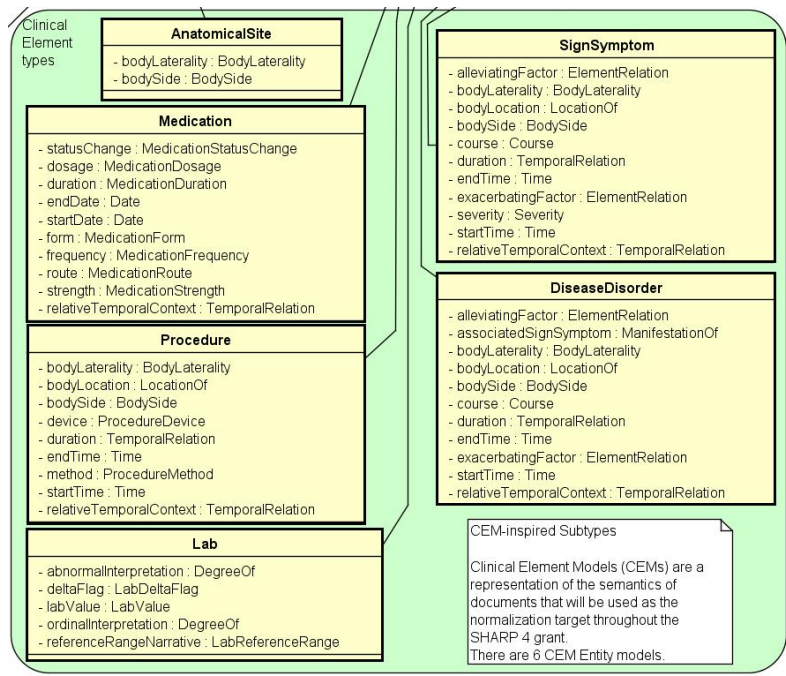


Fig. 1: Core entities in the type system of Apache cTAKES [1].

labor, which is slow and expensive, especially considering the amounts of data needed to train large Transformer-based models.

This paper presents a workflow that can, despite adverse conditions in terms of resources, lead to the creation of annotated health record data of reasonable quality.

2 Preparatory considerations

2.1 Selection of health records for annotation

Corpora of health records often contain distinct categories of medical text ranging from fluent narrative to almost tabular representation of laboratory values. To have enough training material for the dominant text types, but at the same time to be able to cover most of them, a reasonable sampling approach would reflect the ratios present in the whole corpus.

In this project, data for annotation was selected from a corpus of Czech health records collected at the Masaryk Memorial Cancer Institute in Brno, Czech Republic, totaling more than 42 million words in over 150,000 records detailing the stories of more than 4,200 patients. A balanced subset of 168 records, just under 50,000 words, was selected for human annotation.

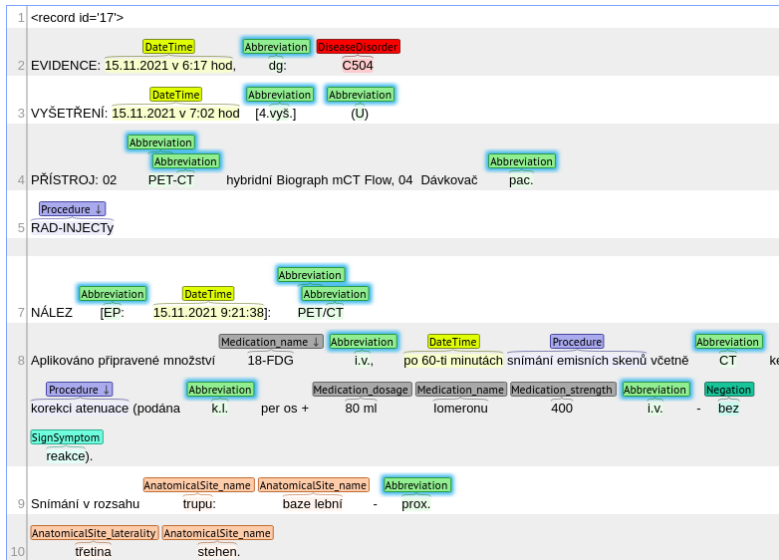


Fig. 2: BRAT tool interface.

2.2 Deidentification of records

Depending on the confidentiality clearance of recruited annotators, deidentification of texts can either be carried out before they are handed to the annotators, or it can be a part of the annotators' task.

In this project, all records selected for annotation were both automatically and manually searched for the occurrence of person names and other identifiers - the students tasked with annotation received safely deidentified data.

2.3 Choice of annotation schema

In order for the results to be commensurable with other research, the chosen annotation schema should be based on a standard already used in the field. This project is based on the six core clinical elements [1] (Figure 1) in the type system of Apache cTAKES [5], a major open-source NLP system for the extraction of clinical information from free text: *AnatomicalSite*, *DiseaseDisorder*, *Lab*, *Medication*, *Procedure*, *SignSymptom*.

To represent a few additional practical categories relevant for the medical domain, *Abbreviation*, *DateTime*, and *Negation* were added (Apache cTAKES represents these in a different way), and several core clinical elements were expanded into multiple annotation types to reflect some of the elements' deeper attributes: *AnatomicalSite_name*, *AnatomicalSite_laterality*, *Lab_name*, *Lab_unit*, *Lab_value*, *Medication_dosage*, *Medication_name*, *Medication_strength*.



Fig. 3: BRAT annotation dialog with the option of entering confidence and abbreviation expansion.

3 Workflow

3.1 Technology

In this project, the BRAT annotation tool [6] was chosen for its lightweight versatility, transparent and reusable file formats, and the option of recording comments and degrees of confidence for each annotation. Figure 2 shows the BRAT annotation interface and Figure 3 shows the annotation dialog box with the options available to annotators.

3.2 Preannotation

To maximize the efficiency of human annotators, any category of medical concepts that can be reliably annotated using rule-based methods should be preannotated before human annotators receive the text. However, these annotations should be editable so that after they are verified by the human, they become authoritative annotations, of a status equal to that of the manually entered ones.

In this project, preannotation vocabularies were compiled for

- names of medications registered in the Czech Republic, using the public database of the State Institute for Drug Control [7]
- common medical abbreviations, merging several available lists [4,2,3]

For an even more thorough preannotation, it is advisable to design regular expressions capturing repetitive character patterns such as

Entities to be annotated

You can view a sample annotation [here](#).

- **AnatomicalSite**
 - names of body parts and locations on the body
 - every **AnatomicalSite** annotation is either of these two:
 - **AnatomicalSite_name**: the name itself, e.g.

našla v pravém **prsu** bulku
 - **AnatomicalSite_laterality**: further specification of location, e.g. v

našla v **pravém** prsu bulku
- **DiseaseDisorder**
 - names of diseases and disorders, e.g.

lčena xareltm **inf mononukleozu** v 15 letech
- **SignSymptom**
 - medical occurrences which are not names of diseases and disorders but can indicate their presence or absence, e.g.

při **bolestech svalů, teplotě**
- **Procedure**
 - If unsure whether it is an official *disorder name* or only a *symptom*, annotate as **SignSymptom**.
 - name of a procedure (diagnostic or therapeutic) carried out by medical personnel, e.g.

benefit **adjuvantní chemoterapie** minimální

Fig. 4: Examples from the annotators' manual.

- quantities and units ("15mm")
- time expressions ("15:22")
- drug dosage regimen ("1-0-1")

and others, if these fit into the chosen annotation schema.

3.3 Human annotation process design

The situation of limited resources often includes the unavailability of experts whose annotation can be considered gold standard without reservation. While it would be enough to have one expert annotate each record, in the more common scenarios where the annotating workforce is only partially qualified or its qualification consisted in a short training, there is reasonable motivation to have multiple annotators per record.

The rationale for this is that multiple-person annotation produces both a high confidence "consensus set" of annotations, where the simultaneous decision of multiple humans to annotate a particular string raises its confidence almost to gold standard level, and also a wide and varied "fuzzy set" of annotations only entered by one of the annotators, which may or may not be perfectly correct, but are still highly valuable for training large language models' entity recognition (after all, automatic medical NER performing as well as a less qualified human annotator would be a grand achievement).

In this project, 11 university students were recruited. Each student received instruction in the form of an annotators' manual (see Figure 4), which explained the technical process of annotating in BRAT and introduced the types of annotations the students were expected to enter. Students were also instructed to revise preannotations and correct them if necessary. Since individual strings can

fall into multiple medical categories, multiple different annotations of the same string were allowed.

5 sub-datasets of just under 10,000 words were created and 2 or 3 annotators were assigned to annotate each of these sub-datasets. To mitigate issues such as failure to complete the task or serial position effect, records were shuffled for each annotator so the starting and ending positions were different for everyone.

Table 1 shows the numbers of annotations acquired in this project while following this workflow.

4 Prospects of LLM training

50,000 words is not enough training material to fine-tune a large Transformer model. But there are multiple options of how it can help create a sufficient amount of data.

4.1 Iterative augmentation

The limited-resource environment dataset of annotations can be used as the basis for a data augmentation process by a series of bootstrapping cycles with the following structure:

1. Training a NER model using the annotated data available in step n
2. Using this model to annotate a larger unannotated dataset planned for step $n+1$
3. Human-reviewing representative amounts of the resulting annotation and tweaking the $n+1$ annotations, e.g. programmatically removing repeated patterns of incorrect annotation
4. Producing a final set of annotations for the $n+1$ dataset
5. Repeating this process with an even larger dataset, using dataset $n+1$ as n

This approach is similar to the work of [9], visualized in Figure 5.

Table 1: Annotation count at different stages of the annotation workflow.

Stage	Annotation count
Initial state of health records	0
Rule-based preannotations	4,266
Preannotations handed to annotators	9,368
New annotations entered by annotators	22,798
Total number of human-verified or human-entered annotations	32,166
Total number of tokens with human-verified or human-entered annotation	45,032

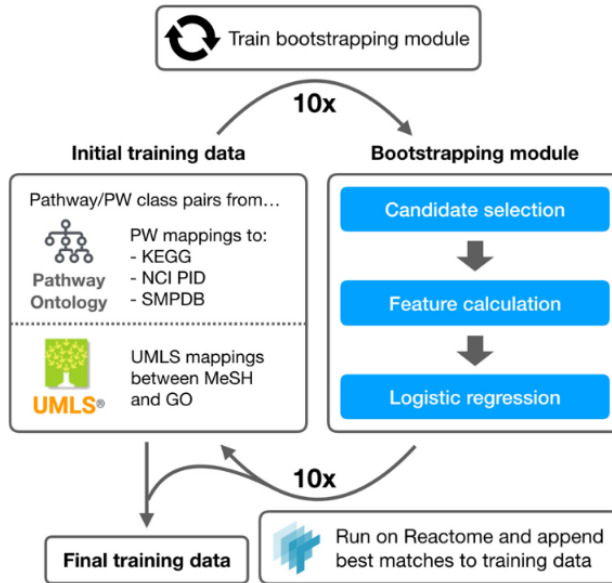


Fig.5: Visualization of the bootstrapping procedure of [9], similar to the one proposed here.

4.2 Data synthesis

Another way of increasing the volume for training large models is to augment the data using a combination of the following:

- Synonym replacement - creating copies of annotated data while replacing human-annotated concepts with different concepts of the same type, e.g. using an external vocabulary of signs and symptoms (UMLS or other) to create variations of original sentences that contain a *SignSymptom* annotation.
- LLM-assisted synthesis of data similar to the original data, varying sentence structures and interchanging entities. This kind of approach recently gained popularity thanks to the fast growth of publicly available large language models. A notable example of a synthetic health record generation approach using prompts for LLMs can be found in [8].

5 Conclusion

It is apparent that this approach introduces many imperfections into the data along the way. However, in limited-resource scenarios, imperfect data is still infinitely better than none. As long as systems trained using such data are used in the capacity of assisting doctors with decisions and making their data more readable, they might easily have a net positive effect, but they first need to be created and evaluated in terms of what they are good for.

This paper serves to demonstrate one of the possible approaches where limited human and data resources are gradually developed into a usable dataset, and to encourage researchers in a similar situation to get inspired by it.

Acknowledgements. The work in this paper was carried out within the project MUNI/G/1763/2020: *AIcope – AI Support for Clinical Oncology and Patient Empowerment*. The analyzed Czech data was kindly provided by the Masaryk Memorial Cancer Institute in Brno, Czech Republic.

References

1. Apache cTAKES - User FAQs — [svn.apache.org](https://svn.apache.org/repos/infra/websites/production/ctakes/content/user-faqs.html). <https://svn.apache.org/repos/infra/websites/production/ctakes/content/user-faqs.html>, [Accessed 09-11-2023]
2. Institute of Endocrinology: List of abbreviations – Institute of Endocrinology — endo.cz (2009), [Accessed 19-10-2023]
3. Karviná-Ráj hospital: List of abbreviations – Karviná-Ráj hospital — nspka.cz (2017), [Accessed 19-10-2023]
4. Jiráková, P.: List of the most common medical abbreviations – Alfabet — alfabet.cz (2014), [Accessed 19-10-2023]
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)
6. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: BRAT: a web-based tool for NLP-assisted text annotation. In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. pp. 102–107 (2012)
7. SÚKL, State Institute for Drug Control: Medicinal Products Database (in Czech Databáze léčivých přípravků DLP) — opendata.sukl.cz (2023), [Accessed 19-10-2023]
8. Tang, R., Han, X., Jiang, X., Hu, X.: Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360* (2023)
9. Wang, L.L., Thomas Hayman, G., Smith, J.R., Tutaj, M., Shimoyama, M.E., Gennari, J.H.: Predicting instances of pathway ontology classes for pathway integration. *Journal of biomedical semantics* **10**, 1–11 (2019)
10. Wornow, M., Xu, Y., Thapa, R., Patel, B., Steinberg, E., Fleming, S., Pfeffer, M.A., Fries, J., Shah, N.H.: The shaky foundations of large language models and foundation models for electronic health records. *npj Digital Medicine* **6**(1), 135 (2023)
11. Yang, X., Chen, A., PourNejatian, N., Shin, H.C., Smith, K.E., Parisien, C., Compas, C., Martin, C., Flores, M.G., Zhang, Y., et al.: Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540* (2022)

Thematic Markers and Keywords on the Example of German Political Discourse

Maria Khokhlova 
and Mikhail Koryshev 

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

Abstract. The paper presents the results of keyword extraction and topic modeling based on LDA model from gensim applied to the texts of a German journal “Merkur” collected from 2017 to 2022. The algorithm extracted the most frequent topics that receive attention in the journal and their change over time. The authors also analyze the similarity of the articles with each other.

Keywords: Topic markers, keywords, German language, political discourse

1 Introduction

Topic modeling, like many other applied tasks, was initially carried out on English data. Each text can be represented by several topics, thus it is possible to determine the similarity of the texts. These topics and related keywords allow one to get an idea of the thematic content of texts and reveal latent semantic structures. Applied to the analysis of political discourse, the selection of thematic markers makes it possible to demonstrate which topics are popular, indicate interest in the author’s position, and give an idea of the main ideas in the text. The paper [6] is devoted to the study of the materials of the US Senate meetings: the selection of keywords, possible topics, their clustering, as well as their change over time. In [3], the texts of speeches at plenary sessions in the European Parliament were analyzed using a non-negative matrix decomposition (NMF). The dynamics of the discussion of various topics from 1999 to 2014 was traced, and a regression model was built that took into account party membership, the number of speeches, voting for or against a party group, etc. Recently, topic models have also been used to cluster literary texts (see, for example, [8]).

In the German linguistic tradition, special attention is paid to key political issues and their reflection in journalism. Attitudes towards the law on renewable energy sources (Erneuerbare-Energien-Gesetz) were studied by the authors on the basis of German newspapers using structural topic modeling (STM) in [2]. The perceptions of the southern countries (Portugal, Italy, Greece, Spain) were

analyzed using the same method in the German-language press from 1946 to 2009 on the material of the newspaper “Die Welt” [4].

Our work is part of a project focused on the study of German and Russian political discourse. The working hypothesis is that the extracted markers make it possible to track changes in topics that receive public interest and values shared by groups of people, outline a range of important issues, as well as attitudes towards them over time, taking into account the historical and cultural context that contributes to these changes. The paper presents the results of applying the procedures for topic modeling applied to the texts of a German journal. The most frequent topics that receive attention in the articles were identified, the change in topics over time was traced, and the similarity of the articles with each other was analyzed.

2 Methodology

2.1 Text selection

Founded as a monthly magazine in 1947, “Merkur” [5] follows the idea proclaimed in the subtitle, i.e. “German magazine about European thinking”: it publishes leading humanists on politics, aesthetics, social studies, economics, art and literature, where questions are raised, currently perceive attention from the professional German university community. These publications are extremely important, as they prepared the transition of the post-war Germany from German-centric thinking in humanitarian higher education to the European and, let us add, transatlantic vision of the post-war period in the spirit of Robert Schumann. A distinctive feature of the published materials is that the articles do not provide a deepening into the particular problems of the narrowly professional occupations of the authors, but still show how the results of particular subject research allow us to come to conclusions and generalizations of an interdisciplinary and generally significant nature, which are important for the professional humanitarian readership as a whole. At the same time, the conclusions of the authors hit the target set out in the subtitle of this publication, serving the cause of the inclusion of German thought in the pan-European and (earlier, transatlantic) now - globalist context. It should be noted that the editors deliberately make the most interesting texts publicly available, thereby expanding their audience and strengthening their influence - it will not be wrong to say that involvement in the texts of “Merkur” is a kind of pass to the German humanities academic world, and the habit of reading and discussing it publications are perceived as a sign of involvement in the current humanitarian agenda.

We selected articles from 2017 to 2022, access to which is carried out without a subscription (this explains the different amount of data, see Table 1), which allows us to look at those central materials for the editorial board, which, taking into account what has been said above should serve to create a common European humanitarian space through the integration into it of German mentality proper.

Table 1: Text data.

	2017	2018	2019	2020	2021	2022
number of texts	54	18	26	33	22	18
number of tokens	99,906	37,519	68,629	131,266	84,040	73,797

In total, the collection comprises 171 texts (about 500 thousand tokens) that prove to be heterogeneous both in their structures and genres.

2.2 Methods

The purpose of our study is to prove, using automatic procedures, the possibility of identification of topics that were manually identified by experts, as well as to describe and assess thematic components of texts from different time periods.

Preprocessing of texts was carried out using the following procedure: lemmatization was carried out using the HanTa [9] tagger (it shows the highest results for German texts), and stop words and auxiliary parts of speech were removed using the NLTK library. Additionally, rare vocabulary and high-frequency words that can “noise” topics were filtered out: words that occur in less than 5 documents and in more than 75% of documents were not considered. Using Latent Dirichlet Allocation (LDA) [1], implemented in the gensim library, seven models were built, which made it possible to identify the most frequent topics for each year and for the entire corpus as a whole.

To select the optimal number of topics, the coherence measure C_v was used, which demonstrates the most successful results in solving this task [7]. The measure takes values from 0 to 1: the higher the value, the greater the coherence between words, respectively, the better the model or the more interpretable the selected topics are.

3 Results

3.1 Coherence

The interval from 4 to 20 topics was chosen as the range of the number of possible topics. Coherence graphs allow determining the optimal number of selected topics for the considered collections of articles (see an example in Fig. 1).

The measure of coherence showed the highest value for the given sample for 10 topics.

In general, the number of topics does not differ significantly for the examined samples (see Table 2), the standard deviation is 1.13. For corpus 2017–2022, as expected, the largest number of topics was proposed ($C_v=0.53$).

The number of selected topics needs to be discussed in detail. If there are few of them, then only general topics will be marked, while with an increase in their number, their “fractionality” increases and more intersections appear, which can make it difficult to interpret the thematic components of the texts.

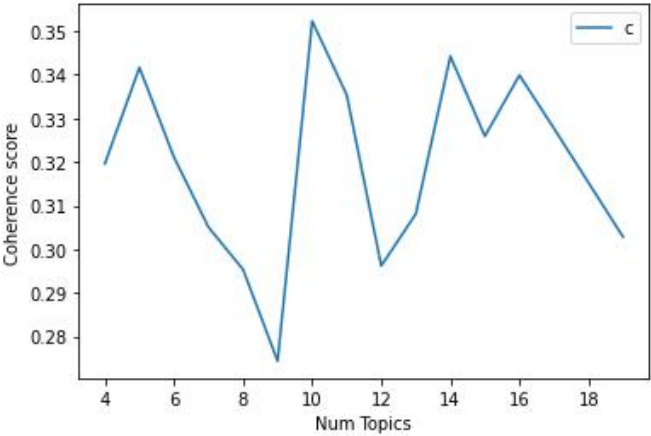


Fig. 1: Coherence for the texts from 2017-2022

Table 2: Number of topics according to the coherence measure.

	2017	2018	2019	2020	2021	2022
number of topics	5	6	6	6	7	8
number of topics ascribed by experts	6	7	4	6	5	5

3.2 Extracted topics

Each topic is presented as a set of keywords with weights assigned to them. In order not to exceed the volume of the paper, we will give an example for one sample of 2017–2022 with the assigned topics:

- 1) books (book reviews, discussion);
- 2) sexism;
- 3) social and political life;
- 4) literary texts;
- 5) literature;
- 6) university life (lectures, freedom of speech);
- 7) culture and art;
- 8) theater (theatrical performances).

Below there is a list of keywords extracted for the topic “university life”:

- (‘Universität’, 0.00974275),
- (‘tun’, 0.009398366),
- (‘Sarrazin’, 0.009035184),
- (‘wissenschaftlich’, 0.007363963),
- (‘Seminar’, 0.0068394854),
- (‘Meinungsfreiheit’, 0.006065996),
- (‘Buch’, 0.00588211),
- (‘deutsch’, 0.005216127),
- (‘ja’, 0.0046629016),
- (‘Du’, 0.00464889),
- (‘Frage’, 0.0044861315),
- (‘Sieg’, 0.0040239994),
- (‘politisch’, 0.003979097),

(‘System’, 0.003978175),
 (‘Wissenschaft’, 0.0038948406),
 (‘sagen’, 0.0037951404),
 (‘Fall’, 0.0037314473),
 (‘schreiben’, 0.0036307815),
 (‘solch’, 0.0036227151),
 (‘Person’, 0.003485797)

Some of the keywords refer to the same topics, showing the intersection between them (Gesellschaft, Kultur). The first topic can be labeled as “writer and creativity” and is the key topic for the June 2019 articles dedicated to the writer Wolfgang Hilbig (*“Den Debilen markieren ... und dann vielleicht klammheimlich schreiben”. Ein Porträt des Arbeiters und Schriftstellers Wolfgang Hilbig*) and to the poet Helen Miles (*“Das Ich ist eine sehr bewegliche Angelegenheit” Interview mit Eileen Myles*). The second theme is, in a sense, a continuation of the first one, but it shows more English words. The next topic is formed by the article *“Die Politisierung der Unpolitischen: Moskau, mein Freund Sergej und das Recht auf Stadt”*, published in September, and deals with political events, protests, as well as human rights.

The LDA algorithm shows the probability with which a document belongs to a certain topic. Thus, the article *“Installation einer Freisprechanlage. Ein vorläufiger Bericht in elf Briefen”* (January 2019) belongs to the first cluster with a probability of 0.99, which includes university-related keywords. The content of the article confirms this: it is devoted to a seminar on philosophy and freedom of speech at universities.

In the case of topic modeling, we are talking about fuzzy clustering: a document can refer to several topics. Table 3 presents quantitative data on the distribution of documents by topic. Texts show three macroclusters: “books”, “sociopolitical life” and “fiction”.

Table 3: Distribution of documents by topics for the corpus 2017–2022.

topic	number of documents
books (book reviews, discussion)	68
sexism	3
social and political life	62
literary texts	21
literature	2
university life (lectures, freedom of speech)	3
culture and art	10
theater (theatrical performances)	2

Figure 2 shows a heat map showing the distribution of the selected topics and corresponding documents. Light colors correspond to a greater likelihood that the document is related to a given topic. The results confirm what was shown

above: it is possible to identify two macro-topics dedicated to fiction and literary studies.

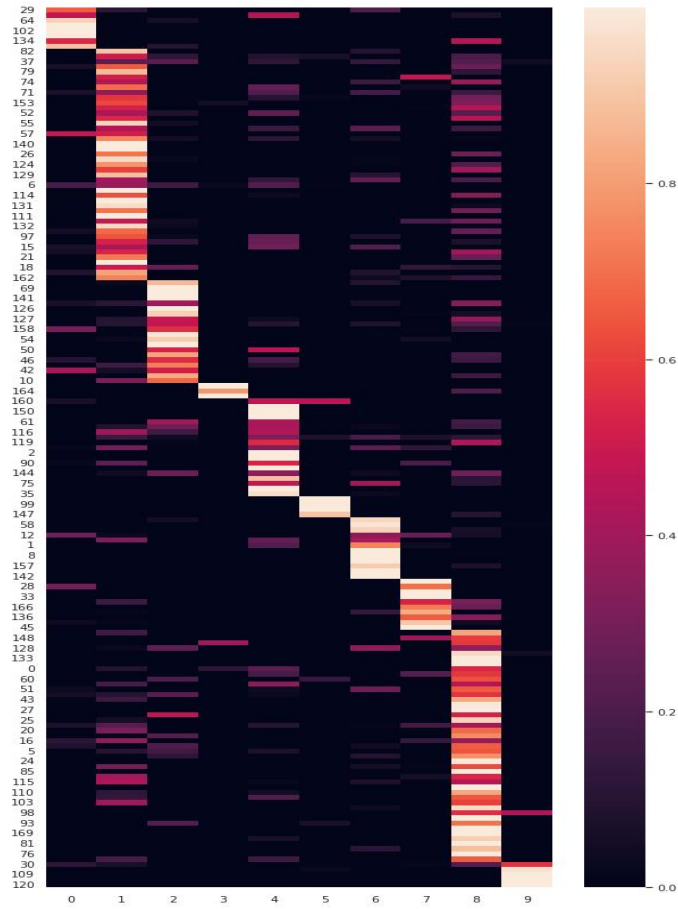


Fig. 2: A heat map for the distribution of documents according to the topics

The last topic, assigned by experts as “fiction” (it reveals general lexis that does not allow building proper clusters for keywords), is controversial, although 18 documents belong to it with a probability of more than 0.4. For example, “Etc. (Warten; Notizen zur leeren Hand)” (September 2018) - an article in which the author reflects on the texts themselves and their comprehension - or “Hausbesuche IV: Bayreuth. Wagner sucht Wagner” (June 2020) that are notes on the Wagner Festival in Bayreuth of the past year. Although issues related to the literature, theater and university life are given a lot of attention on the pages of the magazine, nevertheless, the algorithm attributed only several documents to these clusters.

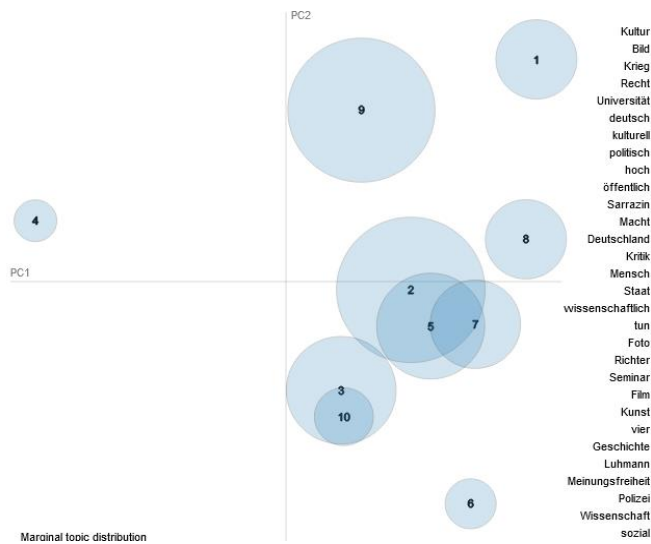


Fig. 3: Intersection of topic clusters

The intersection of topics in clusters is shown in Fig. 3. It should be noted that some topics intersect with each other, which to a certain extent makes it difficult to differentiate them. Topics 1, 4, 8, 6 and 9 are the most distinguishable, as they do not have a common vocabulary among the selected keywords.

The volume of texts in the samples by years is not sufficient: on the example of individual time periods, we noticed that the selected topics are almost identical to the content of some texts and the frequency vocabulary that occurs in them. For example, the article "*Rudolfsheim-Fünfhaus*" (October 2018) with a probability of 0.97 can be assigned a topic related to the description of life in the Rudolfsheim-Fünfhaus area of Vienna, which is very specific.

3.3 Similarity between texts

We calculated the similarity in terms of the cosine measure on tf-idf vectors using the TfidfModel function and the MatrixSimilarity similarity matrix (an example for the articles from 2019 is shown in Fig. 4).

Despite the homogeneous nature of the material, the articles are mostly heterogeneous in their structure and, when compared in pairs, show low similarity. When analyzing texts by years, the greatest similarity (measure value above 0.5) was demonstrated for the following pairs of articles.

1. The topic of elite culture was given attention in a whole series of publications in the magazine, which was reflected in the similarity of the texts:
 - (a) "*Priceless. Die hohe Kultur und das Geld (Hohe Kultur 5)*" (April 2017) and "*Hohe Kultur (7)*" (August 2017).

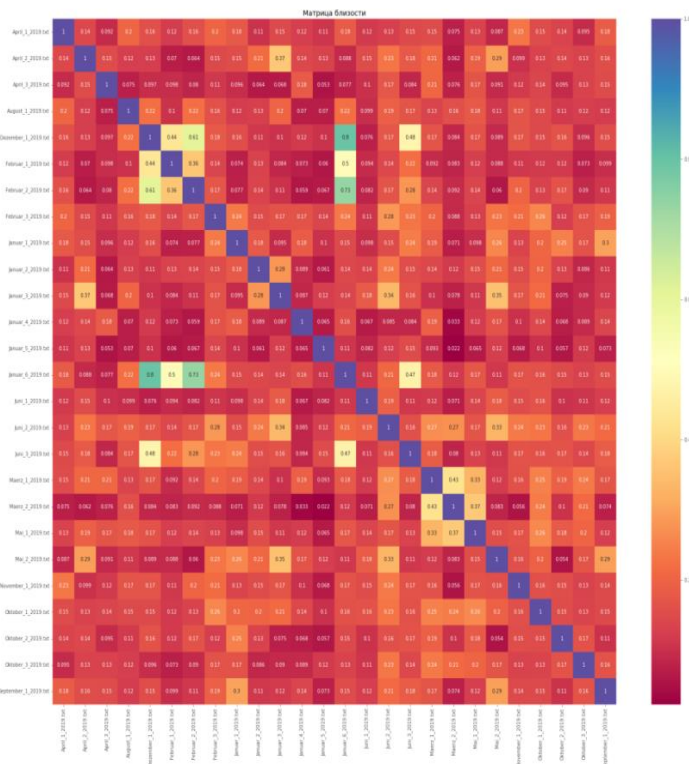


Fig. 4: A heat map for the texts from 2019

- (b) *“Hohe Kultur (7)”* (August 2017) and *“Parteiprogramme: Kulturpolitik (Hohe Kultur 8)”* (September 2017).
- (c) *“Hohe und niedrige Metaphern (Hohe Kultur 2)”* and *“Kooperation Pop und Merkur”* (February 2017).
2. Freedom of speech and academic freedoms was described in the articles:
 - (a) *“Installation einer Freisprechanlage. Ein vorläufiger Bericht in elf Briefen”* (January 2019) and *“Fortsetzung und Abschluss des Berichts: Installation einer Freisprechanlage”* (February 2019). Both texts belong to the same author (Erhard Schüttpelz).
 - (b) *“Installation einer Freisprechanlage. Ein vorläufiger Bericht in elf Briefen”* (January 2019) and *“Wissenschaftsfreiheit und Meinungsfreiheit (Aus Anlass einer Siegener Kontroverse)”* (December 2019). As in the example above, the last article is written by the same author and dwells on the subject of freedom.

3. The lectures of the Swiss writer and journalist K. Kracht:
 - (a) *“Der Autor ist anwesend – Ein Abschlussbericht zu Christian Krachts Frankfurter Poetikvorlesungen”* and *“Blitz und Donner – Christian Krachts Frankfurter Poetikvorlesungen als werkbiographische Zäsur”* (May 2018). These articles are written by the same authors (Kevin Kempke; Miriam Zeh) and are focused on the mentioned topic.
4. The topic of politics and mistakes made in foreign policy issues is described in the articles:
 - (a) *“Über Fehler in der Politik”* (by Ulrich K. Preuß, June 2022) and *“Fehler in der Politik?”*, (by Franziska Davies, September 2022). A later issue provides commentary on the topics that were raised during the summer. Other time periods did not demonstrate similarity (the measure takes a low value of less than 0.4, despite the fact that a number of articles in the journal belong to one topic and are a series of publications). While the articles published in the same year turn out to be thematically similar.

4 Conclusion and Future Work

In the paper, we extracted the most frequent topics for the texts of the articles of the “Merkur” magazine published in different years, as well as evaluated the collection as a whole in terms of its thematic content. The LDA algorithm made it possible to identify significant thematic markers, which generally coincide with the expert evaluation and with the content of the articles. The analysis showed a rather low similarity between the texts of different years, however, within the same year samples, similar texts were identified according to the $tf * idf$ measure.

In general, the algorithm has demonstrated successful results, but additional analysis is needed for such tricky linguistic data. More data is required, as well as the evaluation of topics with keywords identified by other algorithms. It is also important to extract longer n-grams, which will allow for a more “elaborate” identification of topics.

Acknowledgements. The presented research was supported by the Russian Science Foundation, project No. 24-28-00937 “Philological regional studies: mindset of German society in an era of instability”.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (4–5), 993–1022 (2003)
2. Dehler-Holland, J., Schumacher, K., Fichtner, W.: Topic Modeling Uncovers Shifts in Media Framing of the German Renewable Energy Act. *Patterns* 2, 100–169 (2021)
3. Greene, D., Cross, J.P.: Unveiling the Political Agenda of the European Parliament Plenary: A Topical Analysis. In: *Proceedings of the ACM Web Science Conference (WebSci’15)*, Oxford, UK, pp. 1–10 (2015)

4. Küsters, A., Garrido, E.: Mining PIGS. A structural topic model analysis of Southern Europe based on the German newspaper *Die Zeit* (1946–2009). *Journal of Contemporary European Studies* 28 (4), 477–493 (2020)
5. Merkur, <https://www.merkur-zeitschrift.de>. Last accessed 5 Nov 2023
6. Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., Radev, D. R.: How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science* 54(1), 209–228 (2010)
7. Röder, M., Both, A., Hinneburg, A.: Exploring the Space of Topic Coherence Measures. In: *Proceedings of the eight International Conference on Web Search and Data Mining*, Shanghai, February 2–6, pp. 399–408 (2015)
8. Sherstinova, T.Yu., Moskvina, A.D., Kirina, M.A., Karysheva, A.S., Kolpaschikova, A.E.: Thematic modeling of the Russian short story 1900–1930: the most frequent topics and their dynamics [Tematicheskoe modelirovanie russkogo rasskaza 1900–1930: naibolee chastotnye temy i ih dinamika]. In: *Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference “Dialogue 2022”*, Issue 21, vol. 21. Russian State University for the Humanities, pp. 512–526 (2022)
9. Wartena, Ch.: A probabilistic morphology model for German lemmatization. In: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*: Long Papers, Erlangen, Germany, pp. 40–49 (2019)

Subject Index

- AI-detection 35
- annotation 3, 93, 157
- API 11
- authorship identification 57
- automatic dictionary 93

- bilingual lexicon induction 47

- ChatGPT 35
- CIVQA 23
- CIVQA dataset 23, 25
- collocation 79, 121
- concept mining 157
- continuous downloading 107
- corpora 79, 121
- corpus development 107
- corpus processing 107
- cross-lingual word embeddings 47
- Czech 35, 47, 93, 113
- Czechoslovak patterns 113

- database 121
- dataset size 47
- declension 11
- dictionary 93, 101, 121
- dictionary problem 113
- document visual question answering 23

- effectiveness 113
- electronic health records 157
- English 47, 145
- Estonian 47
- evaluation 47, 57, 145

- F0 133

- German 165
- gold standard 121

- hyphenation patterns 113

- keywords 165
- Korean 47

- language model 133, 145
- language services 11
- lemma 93
- lexicon 47
- Lexonomy 101
- log analysis 11

- machine learning 113
- machine translation 145
- manipulative techniques 67
- medical concept 157
- Meligner 133
- Melodic Aligner 133

- Name-Value Hierarchy 101
- named entity recognition 157
- NVH 101

- online behavior 3

- parliamentary protocols 107
- patgen 113
- predictive distributional models 79
- Propaganda dataset 67
- propaganda detection 67

- question answering 23

- reproducibility 57
- Russian 79, 121

- Silesian 145
- Sketch Engine 101
- Slovak 35, 47, 113
- social support 3
- speech melody 133
- speech translation 133
- stylometry 67
- syllabification 113

- tagging 11

text annotation tool	3	word embeddings	47
topic markers	165	XML	101
topics	11, 165		
visual question answering	23	zero-shot	145

Author Index

- Anetta, K. 157
- Denisová, M. 47
- Foltýnek, T. 35
- Horák, A. 67
- Jakubíček, M. 101
- Karásek, A. 57
- Khokhlova, M. 121, 165
- Koryshev, M. 165
- Kovářík, F. 93
- Kovář, V. 101
- Lebedíková, M. 3
- Medved', M. 101
- Mikušek, O. 107
- Mitrofanova, O. 79
- Nevěřilová, Z. 11, 57
- Petrushenko, L. 79
- Plhák, J. 3
- Porteš, D. 133
- Rychlý, P. 47
- Sabol, R. 67
- Ščavnická, Š. 23
- Šigut, P. 35
- Signoroni, E. 145
- Šmahel, D. 3
- Sojka, O. 113
- Sojka, P. 23, 113
- Štefánik, M. 23
- Svoboda, T. 101
- Tkaczyk, M. 3

RASLAN 2023

Seventeenth 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 2023

ISBN 978-80-263-1793-7

ISSN 2336-4289