RASLAN 2024
Recent Advances in Slavonic Natural Language Processing

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

RASLAN 2024

Recent Advances in Slavonic Natural Language Processing

Eighteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2024
Kouty nad Desnou, Czech Republic, December 6–8, 2024
Proceedings

Tribun EU
2024

Proceedings Editors

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

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

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

ISBN 978-80-263-1835-4
ISSN 2336-4289

Preface

This volume contains the Proceedings of the Eighteenth Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2024) , organized by the NLP Consulting, s.r.o. and held on December 6th–8th 2024 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 natural language processing including (but not limited to):

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

RASLAN 2024 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 15 papers were accepted, contributed altogether by 28 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 TEXpertise of Adam Rambousek (based on LATEX macros prepared by Petr Sojka) resulted in the extremely speedy and efficient production of the volume which you are now holding in your hands. Last but not least, the cooperation of Tribun EU as a publisher and printer of these proceedings is gratefully acknowledged.

Brno, December 2024 Aleš Horák

Table of Contents

IV NLP Applications

Part I

Semantics and Language Modelling

From Examples to Patterns: LLM-Generated Regular Expressions for Entity Extraction in Czech Clinical Texts

Petr Zelina

Faculty of Informatics, Masaryk University, Brno, 601 77, Czechia
xzelina2@fi.muni.cz

Abstract. Entity extraction in clinical texts is essential for converting unstructured data in clinical notes into structured formats, facilitating large-scale analysis and clinical decision support. Traditional methods often rely on handcrafted regular expressions (regexes), which, while effective, demand significant time and specialized knowledge to create – resources that healthcare professionals may lack. We introduce a novel approach leveraging large language models (LLMs) to automate regex generation for clinical entity extraction. Our method involves prompting LLMs to generate regex patterns from examples, followed by iterative refinement using a feedback loop. Despite regex limitations, this approach is practical for extracting frequently patterned information common in clinical texts, such as dates, specific data about medical procedures or event detection. Our experiments on Czech clinical notes show this method outperforms current SOTA genetic-programming-based methods for generating regular expression patterns from examples, especially when there are few of them.

Keywords: NLP, LLM, regular expressions, text mining, clinical notes.

1 Introduction

Entity extraction in clinical texts is crucial for transforming unstructured information in clinical notes into structured data, enabling various applications such as clinical decision support, large-scale data analysis or patient cohort selection. One of the traditional methods for extracting entities are handcrafted regular expressions (regexes). Once developed, regular expressions offer several benefits, including speed, portability, and ease of use and deployment. However, creating regex patterns manually is time-consuming and complex, particularly in domains like medicine, where specialized knowledge is essential. Additionally, healthcare professionals may not have the technical expertise required to develop regular expressions, further complicating the manual creation process.

To address these challenges, we propose a novel method that leverages large language models (LLMs) to generate regular expressions automatically. In the most basic form, we show the LLM examples of the clinical extractions and ask it to generate regex patterns that match them. This provides a foundation

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 3–16, 2024.

upon which regex patterns can be iteratively improved by using a feedback mechanism similar to the Tree of Thought approach [10]. Through this iterative process, we expose the model to false positives and false negatives, improving precision and recall. We also branch the LLM conversations into a search tree to leverage the inherent parallelizability of LLMs, improve training speed and stabilize the overall performance of this method.

The limitations of regular expressions mean that this approach is useful for entities that follow a limited number of patterns; however, this is often the case in clinical notes. Examples include extraction of date or time (e.g. for anonymization or event timestamping), searching whether some defined event occurred, or extracting additional information about treatments or procedures missing from structured records.

Recent developments in open large language models (like llama, gemma or mistral) make them usable even for less frequent languages. They enable offline, on-premise use, more easily satisfying privacy requirements.

The source code is available on GitHub[1]

2 Related Work

We can broadly classify approaches for creating regex patterns into three categories – Generating from natural descriptions, generating from in-context examples (annotations) and generating grammars that distinguish between positive and negative examples.

Regex from natural descriptions Kushman and Barzilay (2013) [2] create a syntactic tree of the description and transform it into a regex. Locascio et al. (2016) [5] formulate it as a sequence translation task and apply recurrent neural networks to solve it. However, Zhong et al. (2018a) [13] suggests these approaches fail on real-world datasets.

Zhong et al. (2018b) [12] use expected semantic correctness as the optimization target to mitigate the fact that syntactically different regexes can have the same semantic meaning.

Regex from in-context examples Bartolio et al. (2016) [1] introduce a tool called RegexGenerator++, which uses genetic programming to evolve pattern candidates. As opposed to other methods, it learns from in-context examples, which is exactly the use case for our method as well. It is also one of the few solutions that have complete, working, and easily runnable source code. For these reasons, we compare our method with this one.

Regex from positive and negative matches These methods use a set of positive and a set of negative examples and try to differentiate between them. A prototypical example of this is RegexGolf[2].

[1] `https://github.com/ZepZep/regexulator`

[2] `https://alf.nu/RegexGolf`

Lee et al. (2016) [3] introduce AlphaRegex, which enumerates over possible regexes while using over- and under-approximations to prune the search space.

Li et al. (2020) [4] focus on creating linear-time regex patterns only to avoid Regex Denial of Service attacks.

The fAST approach – Raynal et al. (2023) [7] – infers a grammar based on positive examples only (without context). They offer an easily installable Python module. However, it crashes with an import error as of writing of this paper.

Ye et al. (2020) [11] use both natural descriptions and examples and combine a semantic parser with a program synthesizer. They first let the parser create an incomplete sketch with holes and then use the synthesizer to fill them in so that the pattern agrees with the examples.

3 Proposed Method

The task is to create a regular expression pattern based on example matches.

3.1 Overview

The main idea of the proposed method is to show examples of matches (snippets) to a generative large language model and ask it to develop a regular expression pattern.

To improve this zero-shot scheme, we use an idea similar to Tree of Thought [10] – we iterate on the pattern discovered so far, showing relevant examples (false positives and false negatives) and branching into different paths to achieve more stable performance.

Even though LLM text generation is relatively slow, it can often be efficiently parallelized. We can use this to expand multiple nodes in the branching graph simultaneously. The graph search can be limited by the maximum graph size or a time limit.

To facilitate this, we use two types of nodes – *start* nodes, which are used at the beginning to find initial patterns, and *improve* nodes, which take the current pattern and iterate on it.

We select which nodes to evaluate next based on the validation performance and position in the tree, allowing for more efficient use of resources.

A more rigorous description is available in the following sections.

3.2 Training data

Example set Let E be the Example set. Each example $e_i \in E$ is a pair (d_i, M_i) where d_i is document text and M_i is a set of snippets (matches, extractions). Each $m_{i,j} \in M_i$ is a pair $(l_{i,j}, r_{i,j})$ of indexes into the document d_i, where $l_{i,j}$ points to the first character of the extraction and $r_{i,j}$ points to the last character of the extraction plus one.

Dataset splits In order to improve generalization and maintain good machine learning practices, we can use multiple splits of the example dataset.

- **Training set** T is used during training – the model can use the examples in it directly.
- **Validation set** V is used for comparing different pattern candidates during training. The model does not see its examples directly but has access to the evaluation metrics it provides.
- **Test set** F is used only once after the training to get the final performance for the predicted pattern.

If no validation set is provided, the method uses the training set for both training and comparing nodes.

3.3 Building the exploration tree

We maintain two data structures: a *tree* of nodes, which represents the exploration and dependency of patterns discovered so far, and a priority queue of nodes from the tree that are awaiting evaluation (*task queue*). An example of a small exploration tree is depicted in Figure 1.

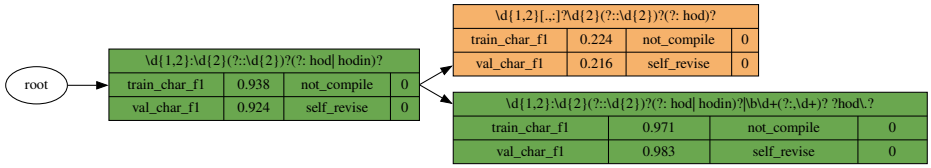


Fig. 1: A small exploration tree with `initial_nodes` = 1, `maximum_depth` = 2 and `branching_factor` = 2. The rectangle records show the resulting pattern of each node, primary training and validation metric and number of internal checks performed. We can see that the bottom *improve node* in the second level led to an improvement, but the top one did not.

We start by initializing several (`initial_splits`) `start` nodes connected to the root of the exploration *tree*. We also add them to the *task queue* with a high priority.

To evaluate nodes from the *task queue*, we use a pool of concurrent workers, which take nodes from the queue and execute them. The execution entails selecting examples to show to the LLM, the conversation with the LLM itself, and extracting the resulting pattern from the LLM's answer. (more details in section 3.4)

After a node is executed, the resulting pattern is evaluated on the validation dataset V. Based on the `branching_factor`, new *improve* nodes are created, initialized with the resulting pattern and appended to the tree as children of the executed node. They are also added to the task queue with priority

$$\texttt{validaton\_score} \times \texttt{depth\_base}^{\texttt{depth}} \times \texttt{sibling\_base}^{\texttt{sibling\_index}}$$

where `depth_base` and `sibling_base` are configurable parameters. The `validation score` can be any metric calculated on the validation dataset. By default it is the mean of `character_f1` and `match_f1` scores (`mean_f1`; see 5.1).

This ensures that the most promising candidates are evaluated first (exploitation) while also maintaining a broader exploration front.

There are two ways for the exploration to stop

- All nodes in the tree are executed. The size of the tree depends on the `initial_nodes`, `maximum_depth` and the `branching_factor` parameters.
- The `time_limit` is reached. From this point, no new nodes start execution, and the currently running ones are awaited.

3.4 Nodes

A node encapsulates one conversation with the LLM. Its execution should result in a new pattern.

Types of nodes We implement two types of nodes

- The *start node* is used at the beginning, where we do not have a candidate pattern. It only shows positive examples to the LLM.
- *improve node* already has a candidate pattern and can use it to select more relevant examples to show – *true positives* (to ensure the new patterns continue to match them), *false negatives* (to show which additional examples should be matched) and *false positives* (to show which examples the pattern currently matches but should not match)

Node workflow The nodes execute the following steps

1. select examples
   - *start node* selects randomly
   - *improve node* selects based on the initial pattern
2. create prompt
3. execute the conversation with the LLM
4. execute internal checks (see below)
5. calculate train metrics

Parts of the prompt Figure 2 shows an example. Definitions of all prompts are available in the code[3]

- Regex Cheatsheet[4] [optional]
- Task description
- Natural description of the matched entity [optional]
- Examples
- Think step by step [9] (encourage the model first to analyze the examples, write down common patterns, and only then create the regex)
- Instructions on how to mark the final regex so it is easy to extract it.

[3] The `prompt_templates.py` file contains all prompt definitions.

[4] `https://regexr.com` → Cheatsheet

```
Write a regular expression that matches all the following parts of
text wrapped in \` and \` as close as possible (but does not have to
be exact). The \` and `\ only highlight the desired extraction in the
examples, they do not appear in the original text and must not be
part of the final regex.

## Examples with added \` \` highlights:
- `12/10/2038`- Mastectomia partialis l. sin. et TAD axillae l
l.sin. Plánování radioterapie, CT simulátor `15.7.2045` v 12.00 hod.

## Only extractions:
12/10/2038
15.7.2045

First, write down common patterns in the extraction. Use them to
construct the regex. Mind the order of the patterns.
Make sure all parentheses match.
Write the final regex on the last line with a heading of FINAL REGEX: ...
Make sure it is on the same line as the heading and is not inside a code
block. Do not write anything else after that.
```

Fig. 2: An example of a *start* prompt with two training snippets (examples data have been altered to preserve privacy)

Formatting the examples The training snippets are shown to the LLM in two ways

- As *highlights with context*. For each example snippet, we create a context string containing it. We use 20–50 context characters, stopping at linebreaks. We use the backtick character to highlight the snippet (the part that should be matched). This character is also used to highlight inline code blocks in the markdown-style format most LLMs use. An alternative could be to use XML tags (for example, `<match>`). Both of these options could suffer if the text of the examples already contains similar markings, so it might be beneficial to choose dynamically.

  In the prompt, it is also important to emphasize that the highlight markings are not part of the text and should not be used in the regular expression.

  This allows the LLM to use context clues, like positive lookaheads, in the regular expression pattern.
- As *extractions*. We also provide the contents of the snippets on their own in the prompt to improve pattern recognition inside the snippet.

The number of training snippets shown at the *start* and during *improvement* is configurable. We use 10 and 5, respectively, as the default.

Internal checks Both node types also come with two internal checks

1. *compile check* – checks if the generated pattern is a valid regular expression. If not, it continues the conversation with the LLM, shows it the compilation error and asks it to provide a fixed pattern.
2. *self validation* – continues the conversation with the LLM and asks it if the provided reasoning makes sense. If not, analyze the error and improve it. It has been shown that self-checking the generated response can improve performance [6].

Both of these checks are optional, and it can be configured how many of them to perform at most.

Node randomness To increase the likelihood that two initial *start nodes* or two *improve nodes* branching from the same pattern come up with different solutions and the branching is not redundant, we provide two ways of *disturbing* the LLM (changing the generated response). First, the sampling and order of examples shown to the LLM are based on a different seed in each node. Second, we can set the LLM sampling temperature to be greater than zero.

4 Experiments

4.1 Datasets

Anonymization in clinical notes The main evaluation dataset we used is a clinical note anonymization dataset [8]. It consists of 80 Czech clinical notes of cancer patients from Masaryk Memorial Cancer Institute. The notes were manually annotated with the label *anonymize*, which should contain all sensitive information. In total, there are 1031 annotations.

To obtain more granular labels, we have classified the extractions into several categories using regular expressions developed in [8] for automatic anonymization. We were left with 157 (15 %) unclassified entries, which we classified manually. Table 1 shows the distribution of labels.

We have chosen to evaluate our method on the *date* and *time* labels as they are suitable for regular expression extraction and have the most annotations.

Table 1: Distribution of labels in the Anonymization dataset [8].

label	date	time	name	hospital	place	job	phone	genetic	multi-date
count	621	146	121	52	40	19	17	8	7

Clinical entity extraction As part of the IDEA4RC[5] project, we are trying to automatically extract different medical entities from clinical notes of cancer

[5] Intelligent ecosystem to improve the governance, the sharing, and the re-use of health data for rare cancers `https://www.idea4rc.eu/`

patients, such as *biopsy type*, *mitotic activity count* or *invasion into fascia*. We are in the middle of an annotation campaign to gather example extractions of these entities. So far, we only have a few (less than 10) examples of each entity, so we cannot offer a full evaluation. Still, we can use this data for qualitative evaluation and discussion.

4.2 Comparison experiment

We compare our approach with RegexGenerator++ [1] as it tackles the same problem and the implementation is available and working. It uses genetic algorithms to find suitable regular expressions.

We create the *evaluation dataset* from the clinical note anonymization dataset (*date* and *time*) with the following steps For each label $l \in \{\texttt{date}, \texttt{time}\}$, for each split $s \in \{1..5\}$ sample half of all examples (notes) as a test set F_s^l. For each number of training examples (notes) $\in \{1, 2, 4, 8, 16, 24\}$ sample a training set $Tl_{s,d}$ from the remaining examples $E \setminus F_s^l$ such that $Tl_{s,d}$ contains d examples (notes) (if possible)

We end up with $2 \times 5 \times 6 = 60$ training sets corresponding to $2 \times 5 = 10$ test sets. The number of snippets (individual matches) in the training sets ranges from 1 to 224.

Configurations

- RegexGenerator++ The main parameters are jobs (j, how many simulations to run), population (p, how many individuals are in each simulation) and generations (g, how many generations each simulation runs). We use three configurations
  - `RG++ small` $j = 8$ $p = 100$ $g = 200$
  - `RG++ medium` $j = 8$ $p = 250$ $g = 500$
  - `RG++ large` $j = 8$ $p = 500$ $g = 1000$
- Our approach. We did not perform any hyperparameter optimizations prior to this test. We do not use `cheatsheet` or `natural_description`.
  The parameters we vary are: number of parallel `workers` w, `initial_splits` i, `max_depth` d, `branching_factor` b, and `time_limit` t (actual time it takes in parenthesis). We use the following parameters
  - `Our` $w = 4$ $i = 4$ $d = 3$ $b = 2$ $t = 60s$ ($80s$)

  As the LLM, we use `llama3.1:70b` available in Ollama with the default `Q4_0` quantization.

Experimental setup The clinical notes contain private information about the patients and doctors. For this reason, we are conducting all our experiments on the Czech CERIT-SC (SensitiveCloud) infrastructure.[6] We have access to an AMD EPYC 9454 CPU and an NVIDIA H100 GPU.

[6] `https://www.cerit-sc.cz/`

We run the RegexGenerator++ with 8 jobs on 8 threads. (more threads would not lead to a speedup as one job cannot run faster, and we already run all of them in parallel)

We use Ollama[7] on the H100 GPU to run the LLMs. We use 4 workers to run 4 conversations in parallel.

5 Results and Discussion

5.1 Metrics

We calculate many different metrics, but the most important are

- **Match F1 score** F1 score for exact matches. Averages precision and recall based on whole predictions. A prediction is classified as true positive only if it matches exactly with an annotation.
- **Character F1 score** F1 score for individual characters. Measures the F1 score as if we classified each character into binary classes. It is more granular than match F1 but may be slightly misleading (e.g. when the model predicts the entire text).
- **Mean F1 score** Arithmetic mean of match F1 and character F1. By default, it is used as the primary metric for selecting the best pattern (on the validation set) and calculating node priorities.

5.2 Comparison experiment

Figures 3 and 4 show the performance of our approach and RegexGenerator++[1]. We show two metrics (`match_f1` and `character_f1`) across two datasets (*time* and *date*). See Section 4.2 for details about the experiments.

Tables 2 and 3 show the means and standard deviations for the *time* dataset.

Table 2: Exact match F1 score ×100 for the *time* dataset. Columns show performance for different numbers of snippets (matches) in the training set. Aggregated from 5 runs.

training snippets	1	2-3	4-7	8-15	16-31	32-63	64-127
RG++ small	00 ± 00	09 ± 17	38 ± 33	76 ± 06	60 ± 22	49 ± 29	35 ± 32
RG++ medium	00 ± 00	33 ± 23	37 ± 32	75 ± 05	59 ± 17	64 ± 14	57 ± 03
RG++ large	00 ± 00	26 ± 18	37 ± 31	78 ± 07	62 ± 20	68 ± 09	60 ± 02
Our	$\mathbf{11} \pm 09$	$\mathbf{45} \pm 12$	$\mathbf{62} \pm 26$	$\mathbf{90} \pm 01$	$\mathbf{89} \pm 03$	$\mathbf{89} \pm 02$	$\mathbf{87} \pm 02$

We can see that our method outperforms RegexGenerator++, especially with low numbers of training snippets. It is also much more stable – the standard

[7] `https://ollama.com/`

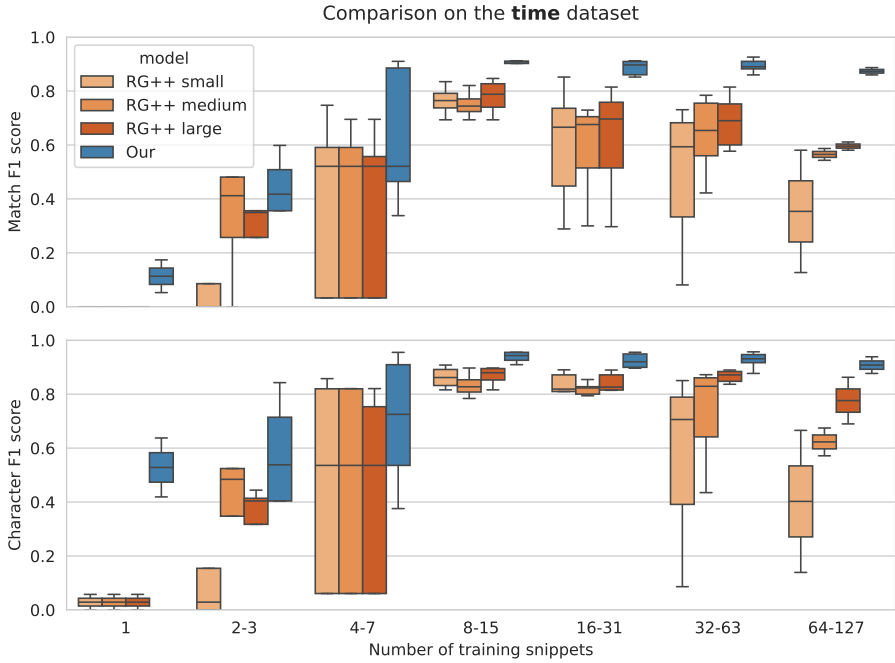


Fig. 3: Comparison of performance on the *time* dataset. The y axis shows two metrics – `match_f1` and `character_f1`. The x axis is stratified based on how many match snippets were in the training set. Aggregated from 5 runs.

Table 3: Character F1 score ×100 for the *time* dataset. Columns show performance for different numbers of snippets (matches) in the training set. Aggregated from 5 runs.

training snippets	1	2-3	4-7	8-15	16-31	32-63	64-127
RG++ small	03 ± 04	13 ± 21	47 ± 39	86 ± 04	79 ± 13	57 ± 34	40 ± 37
RG++ medium	03 ± 04	39 ± 22	46 ± 38	83 ± 05	78 ± 12	73 ± 18	62 ± 07
RG++ large	03 ± 04	33 ± 18	45 ± 37	87 ± 04	81 ± 11	84 ± 07	78 ± 12
Our	$\mathbf{53} \pm 15$	$\mathbf{58} \pm 22$	$\mathbf{70} \pm 25$	$\mathbf{94} \pm 02$	$\mathbf{92} \pm 03$	$\mathbf{93} \pm 03$	$\mathbf{91} \pm 04$

deviation is lower. The instability with fewer training snippets for both methods is caused by insufficient coverage of the training set.

Our method achieves reasonable performance with as few as 4–7 training snippets. If an exact match is not required and the approximate location of the extracted entity is sufficient, the method starts to work even with 1-3 training snippets.

Table 4 shows how long it takes to calculate the predictions. We can see that the speed of RegexGenerator++ is dependent on the number of training examples, as it is used to calculate fitness for each individual many times.

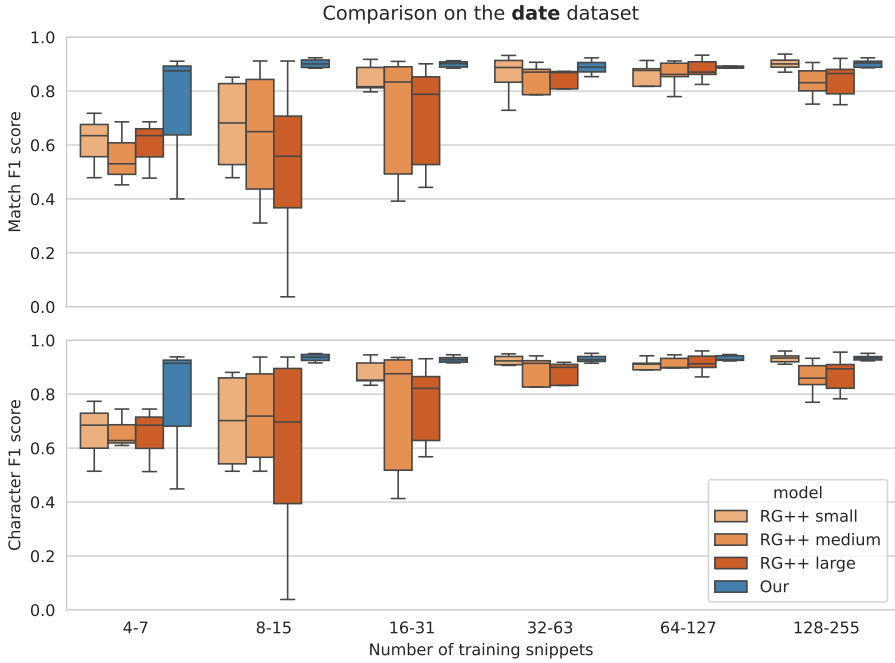


Fig. 4: Comparison of performance on the *date* dataset. The y axis shows two metrics – `match_f1` and `character_f1`. The x axis is stratified based on how many match snippets were in the training set. Aggregated from 5 runs.

Our method shows a constant number of snippets to the LLM in each node and spends almost all the time waiting for the LLM. In this configuration, the `time_limit` was set to 60 seconds. One LLM conversation with `llama3.1:70b` on the H100 usually takes 15–30 seconds. All running conversations are awaited and finished after the `time_limit` runs out, which explains the increased time. As we perform 4 conversations in parallel, each run manages to finish around 16 conversations.

Table 4: Average training time in minutes based on number of training snippets.

training snippets	1	2-3	4-7	8-15	16-31	32-63	64-127	128-255
RG++ small	0.1	0.1	0.2	0.4	1.0	2.3	2.4	1.6
RG++ medium	0.7	0.8	0.8	2.5	4.0	9.2	10.7	9.2
RG++ large	3.0	2.5	5.4	10.3	11.8	38.9	37.9	42.9
Our	1.4	1.2	1.2	1.3	1.3	1.3	1.3	1.4

5.3 Clinical entity extraction

We have access to prototype annotations of several entities in clinical notes that need to be extracted to facilitate federated large-scale data analysis in the IDEA4RC project[8]. As there are only a few annotations for each entity so far, there is not enough data to rigorously measure the performance. However, we offer a qualitative analysis of the behaviour of our approach on this data.

– *type of biopsy*, *invasion into fascia*, *type of necrosis*
  Our approach is able to join different formulations into a single pattern. It is also able to use positive/negative lookaheads.

  Pattern generated for *type of biopsy*:

```
(punkční biopsii|Excidát|resekát|Exstirpát|Excize)|
biopsie|Resekát|Resekce|excize|biospie(?= z| č\.| \d+)
(?!biopsie|resekát)
```

  Pattern generated for *type of necrosis*:

```
nekrotick[ý|é]|nekrotické|fokálně nekrotick[ý|é]|
nekróz(y|ami)?|Bez (ložisek )?nekros
```

– *tumour size*
  Our approach successfully created a pattern for matching dimensions (e.g. `99x99x99 mm`). However, there are many different places where dimensions are used, and the desired extractions were only the primary tumour dimensions. More training examples would be needed to learn the context.

  Pattern generated for *tumour size*:

```
\d+(?:\.\d+)?(?:×|x)\d+(?:\.\d+)? ?mm
```

– *mitotic activity count*
  Here, we have only 3 examples, each in a different format:
  `32 mf / 1,7 mm2` `45/10 HPF` `7-10/10HPF`
  The model matched two of them (the two `HPF`). The first (`mf/mm2`) was not matched only because the regex expects a decimal dot instead of a decimal comma.

  Pattern generated for *mitotic activity count*:

```
\d+(?:-\d+\/\d+HPF|\s*\/\s*\d+\s*HPF|
\s*mf\s*\/\s*[0-9]+(?:\.[0-9]+)?\s*mm2)
```

[8] Intelligent ecosystem to improve the governance, the sharing, and the re-use of health data for rare cancers `https://www.idea4rc.eu/`

6 Conclusion

We have introduced a new approach for generating regular expression patterns based on in-context examples using generative large language models (LLMs). We validate the effectiveness on date and time extractions from clinical notes and compare it with RegexGenerator++[1]. Our experiments show that our approach achieves better results with fewer examples and is more stable.

Finally, we conduct a qualitative study of performance on a small clinical entity extraction dataset. The results seem promising, as the model can merge different static patterns (without wildcards), use positive and negative lookaheads, and handle even more complex patterns with wildcards.

Future work We want to focus on generalizing the approach to be able to handle both positive and negative context-free examples and natural descriptions. Then, we want to compare it with other existing methods in these settings.

We also want to investigate the effects of different hyperparameters, like the LLM model, using the cheatsheet, including natural description, size of the search tree, number of snippets shown or the size of snippet context. For this, however, we will need a harder dataset so that it can properly differentiate the effects.

Finally, once we have access to full annotations from IDEA4RC, we want to do a proper evaluation on this data.

Acknowledgements. Supported by the project SALVAGE (OP JAK; reg. no. CZ.02.01.01/00/22_008/0004644) – co-funded by the European Union and by the State Budget of the Czech Republic.

Supported by grant MUNI/A/1590/2023: Using artificial intelligence techniques for data processing, complex analysis and visualization of large-scale data (AI-for-data)

Supported by the Technology Agency of the Czech Republic project TQ12000018.

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.

Disclosure of Interests. No competing interests to declare.

References

1. Bartoli, A., Lorenzo, A.D., Medvet, E., Tarlao, F.: Inference of regular expressions for text extraction from examples. IEEE Transactions on Knowledge and Data Engineering **28**(5), 1217–1230 (May 2016). https://doi.org/10.1109/TKDE.2016.2515587

2. Kushman, N., Barzilay, R.: Using semantic unification to generate regular expressions from natural language. In: Vanderwende, L., Daumé III, H., Kirchhoff, K. (eds.) Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 826–836. Association for Computational Linguistics, Atlanta, Georgia (Jun 2013), https://aclanthology.org/N13-1103
3. Lee, M., So, S., Oh, H.: Synthesizing regular expressions from examples for introductory automata assignments. SIGPLAN Not. **52**(3), 70–80 (Oct 2016). https://doi.org/10.1145/3093335.2993244
4. Li, Y., Xu, Z., Cao, J., Chen, H., Ge, T., Cheung, S.C., Zhao, H.: Flashregex: Deducing anti-redos regexes from examples. In: 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 659–671 (2020)
5. Locascio, N., Narasimhan, K., DeLeon, E., Kushman, N., Barzilay, R.: Neural generation of regular expressions from natural language with minimal domain knowledge. In: Su, J., Duh, K., Carreras, X. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1918–1923. ACL, Austin, Texas (Nov 2016). https://doi.org/10.18653/v1/D16-1197
6. Miao, N., Teh, Y.W., Rainforth, T.: Selfcheck: Using llms to zero-shot check their own step-by-step reasoning (2023), https://arxiv.org/abs/2308.00436
7. Raynal, M., Buob, M.O., Quénot, G.: fast: regular expression inference from positive examples using abstract syntax trees. In: Coste, F., Ouardi, F., Rabusseau, G. (eds.) Proceedings of 16th edition of the International Conference on Grammatical Inference. Proceedings of Machine Learning Research, vol. 217, pp. 96–116. PMLR (10–13 Jul 2023), https://proceedings.mlr.press/v217/raynal23a.html
8. Rusnačková, K.: Anonymisation of clinical notes (2024), https://is.muni.cz/th/rf3hj/
9. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models (2023), https://arxiv.org/abs/2201.11903
10. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models (2023), https://arxiv.org/abs/2305.10601
11. Ye, X., Chen, Q., Wang, X., Dillig, I., Durrett, G.: Sketch-driven regular expression generation from natural language and examples (2020), https://arxiv.org/abs/1908.05848
12. Zhong, Z., Guo, J., Yang, W., Peng, J., Xie, T., Lou, J.G., Liu, T., Zhang, D.: SemRegex: A semantics-based approach for generating regular expressions from natural language specifications. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1608–1618. ACL, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1189
13. Zhong, Z., Guo, J., Yang, W., Xie, T., Lou, J.G., Liu, T., Zhang, D.: Generating regular expressions from natural language specifications: Are we there yet? In: AAAI Workshops (2018), https://api.semanticscholar.org/CorpusID:20221107

Negation Disrupts Compositionality in Language Models
The Czech Usecase

Tereza Vrabcová and Petr Sojka

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

Abstract. In most Slavic languages, the negation is expressed by short "ne" tokens that do not affect discrete change in the meaning learned distributionally by language models. It manifests in many problems, such as Natural Language Inference (NLI).
We have created a new dataset from CsFEVER, the Czech factuality dataset, by extending it with negated versions of hypotheses present in the dataset. We used this new dataset to evaluate publicly available language models and study the impact of negation on the NLI problems.

We have confirmed that compositionally computed representation of negation in transformers causes misunderstanding problems in Slavic languages such as Czech: The reasoning is flawed more often when the information is expressed using negation than when it is expressed positively without it.
Our findings highlight the limitations of current transformer models in handling negation cues in Czech, emphasizing the need for further improvements to enhance language models' understanding of Slavic languages.

Keywords: negation, language models, machine learning.

"Negation is the mind's first freedom, yet a negative habit is fruitful only so long as we exert ourselves to overcome it, adapt it to our needs; once acquired it can imprison us."
Emil Cioran [1, page 207]

1 Motivation

Truthfulness matters. Handling the representation of sentences with negation has always been a challenge. [4] The compositional nature of languages causes problems with negation in the latest language models (LM) with transformer architecture. Recent studies confirm that even large LMs trained to represent the meaning expressed in sentences with or without negation are biased [10].

The impact of negation on LM tasks is most studied within the sphere of English-based natural language processing. One of the more straightforward

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 17–24, 2024.

methods to evaluate the impact is the cloze task, wherein the input sentence has one or more tokens masked, and the model predicts the missing tokens. This task is highly unconstrained, with many possible correct answers.

Multiple studies [3,6,10] inspect the model's understanding of negation by comparing the model's completions between sentence pairs with opposite polarities. Models commonly correctly complete sentences with positive polarity but generate factually incorrect continuations for the negative counterparts. Notably, the models often suggest the same prediction for both, as is illustrated in Table 1.

Table 1: Example of the LM prediction from the paper [3]. The model is insensitive to the presence of negation, which causes factually incorrect prediction.

Input Sentence	**Correct Token**	**Predicted Token**
A sparrow is a <MASK>.	bird	bird
A sparrow is not a <MASK>.	tree	bird

Although in English, this methodology showcases the model's insensitivity to negation, the expressive power of the cloze task in Slavic languages is limited. From the morphological standpoint, English belongs to the family of *analytic languages*; that is, the syntactic relations of a sentence are expressed using specific grammatical words and a rather fixed word order. Most Slavic languages, on the other hand, belong to the family of *fusional languages*, capturing the syntactic relations with affixes, allowing for looser word order. This results in the already highly unconstrained task gaining even more complexity for evaluation and the lowering of the changes for any possible meaningful value of the results.

In this paper, we evaluate the 'negation bias' in the Czech language and confirm it using several multilingual language models.

2 Methodology

Natural Language Inference (NLI) [7] is a valuable task for evaluating models' understanding of negation. In this task, the model is given a text pair. One of the texts, commonly referred to as a *premise*, contains a knowledge base that is to be taken as truthful. The model is then evaluated based on its ability to correctly determine whether or not the other text from the pair, commonly known as the *hypothesis*, is supported by the information presented in the premise. The hypothesis is then classified as one of three categories based on its textual entailment: it is either supported by the premise, refuted by it, or there is not enough information.

In our experiments, we aim to evaluate the sensitivity of the models' accuracy based on the *presence of negation* in the hypothesis.

We create paired evaluation data – each example comes in two variants. Both variants share the same premise, and the hypotheses are negations of each other, one being supported by the premise and the other refuted by it.

```
[
    {
       "role": "system",
     "content": "You are a fact checker for queries in the Czech language.
     You will be given a premise, which you know is factually correct, and
     a hypothesis. You will return the truth value of the hypothesis, based
     on the premise. Return True if the hypothesis is correct and False if
     the hypothesis is incorrect."
    },
    {
      "role": "user",
      "content": [
        "Premise: Antigua a Barbuda. Jméno zemi dal Kryštof Kolumbus
        v roce 1493 po objevení ostrova na počest Panny Marie La Antigua
        v sevillské katedrále.",
        "Antigua a Barbuda nebyla rodištěm Kryštofa Kolumba."
      ]
    }
]
```

Fig. 1: Example of the evaluation prompt with the system message, containing the task explanation, and the user message, containing the premise and the hypothesis for evaluation.

If the model processes negations correctly, its accuracy should remain the same regardless of whether the hypothesis contains negation or not.

3 Data

Our aim was to evaluate the language model's ability to correctly determine which sentence of the provided hypothesis pair is correct given the specified premise. The first step was the location of a suitable test dataset. There is a couple of Czech NLI test datasets, for our experiments we have used a modified version of the CsFEVER dataset [11].

Each entry in the original CsFEVER dataset contains the following:

- entry id,
- label of the textual entailment category,
- hypothesis (called claim in this dataset), and
- premise (called evidence in this dataset).

First, we removed any entries with the "Not enough information" label and entries containing empty values. Second, we have created a pipeline to generate hypothesis pairs for each premise.

3.1 Sentence Negation Pipeline

As there are many ways to express negation in the Czech language, with varying levels of complexity, we have focused on the method of expressing syntactic negation using a negative morpheme (prefix "ne") to negate the verb within the hypotheses.

The pipeline consists of the following:

1. The hypothesis is first tagged using the UDPipe service [9], a pipeline for tokenization, tagging, and lemmatization. We have used the current Czech model, `czech-pdt-ud-2.12-230717`.
2. We extract the first verb in the hypothesis – a word with either the VERB or AUX tag. AUX stands for the auxiliary verb, in Czech it is the verb "to be".
3. We analyse the extracted verb using the morphological analyser Majka [8], obtaining the verb's lemma and morphological tags. We filter the results to include only verbs to avoid picking the wrong homograph.
4. We modify the tags to reverse the polarity of the verb.
5. We generate the negated version of the verb based on the lemma and the modified tags. We return the verb with the value with the value of the original polarity.
6. We create a copy of the original hypothesis, replacing the verb with its negated version.
7. Based on the textual entailment label and the value of the original polarity, we determine which hypothesis is the truthful one.

The created dataset was named CsNoFEVER and is structured as described on Figure 2.

The naïve implementation of the sentence negation pipeline necessitated manual fine-tuning of the dataset – for hypotheses containing general or strong negation, multiple words beside the verb are impacted and need to be modified. The pipeline also removes non-parsable sentences from the final dataset. The final dataset currently consists of 2,600 entries, in comparison to the original number of 10,000 entries. The final dataset CsNoFEVER is available in the GitHub repository [12].

4 Results

Using CsNoFEVER dataset specified in Section 3 and the methodology specified in Section 2, we have evaluated the NLI task on language models listed in Table 2.

Table 3 summarizes our main results. We show that the inclusion of negation in the evaluated hypotheses has a definite negative impact on the models'

```
{
  "id": 10567,
  "premise": "Google Search, běžně označovaný jako Google Web Search nebo
  jednoduše Google, je internetový vyhledávač vyvinutý společností Google.
  Patří sem synonyma, předpovědi počasí, časová pásma, burzovní kotace,
  mapy, údaje o zemětřesení, časy promítání filmů, letiště, seznamy domů
  a sportovní výsledky.",
  "positive_hypothesis": "Vyhledávač Google zobrazuje informace o domovech.",
  "negative_hypothesis": "Vyhledávač Google nezobrazuje žádné informace o
  domovech.",
  "correct_polarity": "P"
}
```


Fig. 2: Structure and contents of an entry of the CsNoFEVER dataset: `id` of the entry in the original CsFEVER dataset, premise `premise`, pair of hypotheses `positive_hypothesis` (P), `negative_hypothesis` (N), and `correct_polarity` of the hypothesis ("P" or "N").

Table 2: Attributes of language models evaluated during the experimentation. All models (*Mistral-Nemo*, *Qwen2.5*, and *Llama-3.1*) are open-source and downloadable from the HuggingFace website [5].

Model Name	Size	Layers	Languages	Developer(s)
`mistralai/Mistral-Nemo-Instruct-2407`	12.20 B	40	11+	Mistral AI
`Qwen/Qwen2.5-7B-Instruct`	7.62 B	28	29+	Alibaba Cloud
`meta-llama/Llama-3.1-8B-Instruct`	8.06 B	32	8+	Meta

accuracy, with different degrees of severity. It confirms that the study of Truong et al. [10] also holds for Slavic languages such as Czech, where it exemplifies even more than in the languages with the fixed word order.

In the paper's Appendix A, we list several examples of model reasoning and errors they make. It is yet to be investigated to which extent the reasons for errors are primarily due to architectural constraints of transformer architecture such as compositionality of token processing, model size, number of layers, balancing of languages used for training, the models' training parameters, or just because of suboptimal prompting.

Table 3: Model accuracy (P = positive hypotheses, N = negative hypotheses).

Model Name	P Accuracy	N Accuracy
Mistral-Nemo	79.81% (2075)	38.73% (1007)
Qwen2.5	87.38% (2272)	76.15% (1980)
Llama-3.1	71.35% (1855)	51.35% (1855)

"Knowledge is two-fold, and consists not only in an affirmation of what is true, but in the negation of that which is false." Charles Caleb Colton [2]

5 Conclusion and Future Work

In this paper, we have investigated the difficulties language models have in correctly carrying out the NLI task on texts that include negation. We have created a new dataset for our evaluation, CsNoFEVER, modifying the CsFEVER test dataset and expanding it to contain more instances of negative hypotheses. Evaluating a number of language models, we have shown that even the latest language models, within the tested size range, have trouble correctly classifying texts with negation.

Due to the simplicity of the sentence negation pipeline introduced in this paper, the CsNoFEVER currently does not contain complex expressions of negation. We plan to further develop the negation pipeline, allowing it to parse and generate more complex cases of negation for future iterations of the dataset CsNoFEVER.

References

1. Cioran, E.: The Temptation to Exist. Quadrangle Books (1956), https://dl1.cuni.cz/pluginfile.php/1176322/mod_resource/content/1/Cioran_Temptation.pdf
2. Colton, C.C.: Lacon, or Many Things in Few Words. London: Longman, Hurst, Rees, Orme and Brown (1820)
3. Ettinger, A.: What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. Transactions of the Association for Computational Linguistics **8**, 34–48 (Jan 2020). https://doi.org/10.1162/tacl_a_00298
4. Hermann, K.M., Grefenstette, E., Blunsom, P.: "Not not bad" is not "bad": A distributional account of negation. In: Allauzen, A., Larochelle, H., Manning, C., Socher, R. (eds.) Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality. pp. 74–82. ACL, Sofia, Bulgaria (Aug 2013), https://aclanthology.org/W13-3209
5. Hugging Face – The AI community building the future, https://huggingface.co/
6. Kassner, N., Schütze, H.: Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In: Proceedings of the 58th Annual Meeting of the ACL. ACL (2020). https://doi.org/10.18653/v1/2020.acl-main.698
7. Putra, I.M.S., Siahaan, D., Saikhu, A.: Recognizing textual entailment: A review of resources, approaches, applications, and challenges. ICT Express **10**(1), 132–155 (2024). https://doi.org/https://doi.org/10.1016/j.icte.2023.08.012
8. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2009. pp. 13–16 (2009), https://nlp.fi.muni.cz/raslan/2009/papers/13.pdf
9. Straka, M., Straková, J.: Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Hajič, J., Zeman, D. (eds.) Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 88–99. Association for Computational Linguistics, Vancouver, Canada (Aug 2017). https://doi.org/10.18653/v1/K17-3009

10. Truong, T.H., Baldwin, T., Verspoor, K., Cohn, T.: Language models are not naysayers: an analysis of language models on negation benchmarks. In: Palmer, A., Camacho-collados, J. (eds.) Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023). pp. 101–114. ACL, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.starsem-1.10
11. Ullrich, H., Drchal, J., Rýpar, M., Vincourová, H., Moravec, V.: Csfever and ctkfacts: acquiring czech data for fact verification. Lang. Resour. Eval. **57**(4), 1571–1605 (May 2023). https://doi.org/10.1007/s10579-023-09654-3
12. Vrabcová, T.: CsNoFEVER: Negated CsFEVER dataset (2024), `https://github.com/xvrabcov/cs-no-fever`

A Examples of Models' Reasoning

This appendix contains examples showing the output and reasoning of the tested models.

A.1 Example: Scandinavian Mountain Range

Premise: Pohoří Rincon (o'odhamsky: Cew Do'ag) je významné pohoří východně od Tucsonu v okrese Pima v Arizoně ve Spojených státech. Národní park Skalisté hory (anglicky Rocky Mountain National Park) je národní park ve Spojených státech amerických, který se nachází přibližně 76 km severozápadně od mezinárodního letiště v Denveru v severní části centrálního Colorada, v pohoří Front Range of the Rocky Mountains. Mezi hlavní rysy parku patří hory, vysokohorská jezera a široká škála divoké zvěře v různých klimatických podmínkách a prostředích, od zalesněných lesů po horskou tundru. Skandinávie. Velká část skandinávských hor má klima alpínské tundry.
Positive Hypothesis (“P”): Skandinávie zahrnuje pohoří.
Negative Hypothesis (“N”): Skandinávie nezahrnuje pohoří.
Correct Polarity: “P”
Model Predictions:
Mistral-Nemo: “P”. Pohoří Rincon je významné pohoří východně od Tucsonu v okrese Pima v Arizoně ve Spojených státech, a pohoří Skandinávie je v pohoří Front Range of the Rocky Mountains, a je to významné pohoří.
Qwen2.5: “N”. Skandinávie se jedná o oblast s vlastními horami, jako jsou fjords a skandinávské Alpy, a tedy nezahrnuje pohoří.
Llama-3.1: “N”. Skandinávie je geografický region, který zahrnuje země jako Norsko, Švédsko a Finsko, a podle textu se v něm nachází alpínská tundra, nikoli pohoří.

A.2 Example: Canadian Rapper

Premise: One Dance je píseň kanadského rappera Drakea z jeho čtvrtého studiového alba Views (2016).
Negative Hypothesis (“P”): Autorem skladby One Dance je Američan.

Negative Hypothesis (“N”): Autorem skladby One Dance není Američan.
Correct Polarity: “N”
Model Predictions:
Mistral-Nemo: “P”. Píseň ’One Dance’ nenapsal Drake, vzhledem k tomu, že skladbu napsal kanadský rapper Drake.
Qwen2.5: “N”. Autor skladby One Dance, Drake, je kanadský rapper, ne americký.
Llama-3.1: “P”. Hypotéza P je správná, protože Drake je kanadský rapper a kanadský občan.

A.3 Example: Area of Yemen

Premise: Jemen je druhou největší zemí na poloostrově, zabírá 527 970 km2 (203 850 km²).
Negative Hypothesis (“P”): Jemen má rozlohu 250 750 km².
Negative Hypothesis (“N”): Jemen nemá rozlohu 250 750 km².
Correct Polarity: “N”
Model Predictions:
Mistral-Nemo: “P”. Jmenovitý údaj uvádí, že rozloha Jemenu je 203 850 km².
Qwen2.5: “N”. Podle vstupního předvolbu má Jemen 527 970 km2 rozlohy, nikoliv 203 850 km2.
Llama-3.1: “N”. Hypotéza P je nesprávná, protože uvádí rozlohu Jemenu 250 750 km², zatímco v předloze je uvedena rozloha 527 970 km².

SlamaTrain – Representative Training Dataset for Slavonic Large Language Models

Marek Medved'[1,2], Radoslav Sabol[1], and Aleš Horák[1]

[1] Masaryk University, Faculty of Informatics
Brno, Czechia,
[2] Lexical Computing
xmedved1@fi.muni.cz, xsabol.fi.muni.cz, hales.fi.muni.cz

Abstract. The Slama project focuses on building a series of foundational language models for Slavonic languages. Even though the latest development yields a number of new large pre-trained and fine-tuned models, the main data source came from English-written websites. Therefore the majority of the training data that is used for language model development consists of the English language. Multilingual language models like Llama, GPT-4o, mT5, etc. are also predominantly (around 80%) trained on the English language, even though they capture the structure of dozens of languages.
In this paper, we detail the process of acquiring one of the largest training datasets for Czech, Slovak and other Slavonic languages. We started with huge multi-lingual datasets, extracted the mono-lingual data and joined them with other sources. The combined mono-lingual datasets were then cleaned, deduplicated and filtered for adult content. As a result, we have obtained 71 billion tokens for the Czech and Slovak languages suitable for the Slama language models training.

Keywords: Slama models, LLM, large language models, training, dataset

1 Introduction

Large language models (LLMs) require extensive and good-quality collections of texts for the causal language modeling task. First LLMs, such as BERT [3] or RoBERTa [9] have been trained purely on English collections of Wikipedia articles or book texts. Current LLMs, such as Meta Llama [6] or Mistral Nemo [10] are built from several orders of magnitude larger networks and for maximum exploitation of the encoded model "memory", they need appropriately larger text collections in tens of languages. However, most of the largest collections are based on Common Crawl [12] that, naturally reflecting the proportions of texts available on-line, are imbalanced in favor of English and other mainstream languages.

A multilingual LLM can process input in most of the languages contained in its training data, however, its internal semantic representations incline to follow

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 25–33, 2024.

the word and phrase senses of the biggest languages. In the following sections, we present SlamaTrain,[3] a new representative training dataset for Czech, Slovak and other Slavonic languages.

2 The Dataset Building Process

The Slama dataset focuses on combining large language datasets into one source that is more focused towards the Slavonic language family. This requirement affects the dataset in favor of Slavic languages, which make up the core of the Slama dataset. The language proportions of the dataset are thus intentionally skewed so that the standard mainstream languages do not outnumber the Slavonic parts. Furthermore, the processing pipeline ensures data de-duplication, non-Latin-script removal and adult content filtering to prevent Large Language models from generating unwanted texts.

2.1 Sources

The Slama dataset consists of Czech, Slovak, Polish, Slovene, Croatian, English, French, Italian, German and Spanish texts. Even though the Slama project primarily focuses on Slavonic languages the dataset also contains selected European mainstream languages that are frequently used (English, German, etc.) and present in Slavonic texts, however, their proportions are reduced.

The texts of the Slama dataset come form several sources:

- CulruraX dataset[4] [11]: this multilingual dataset consisting of 6.3 trillion tokens and 167 languages was developed for training LLM models. Due to its size, data cleanliness that undergoes multiple stages and MinHash at document level de-duplication, it constitutes the main part of Slama dataset.
- HPLT dataset [7]: for the languages in focus (Czech, Slovak, etc.), the HPLT 1.2 dataset served as the second big source. The monolingual part of HPLT includes 75 languages in this release which resolved in 11 TB of de-duplicated files and 8.4 TB of clean files. In the Slama dataset, we include the Czech, Slovak and Slovene part of HPLT 1.2 dataset.
- TenTen corpora [8]: is a family of web corpora created by the Sketch Engine team. Each monolingual corpus usually contains at least 1 billion tokens in the target language. The Czech and Slovak TenTen corpora are included in the Slama dataset.
- Aranea corpora [2]: is a family of Slovak-centric corpora, de-duplicated on the document level and harvested from the web using the SpiderLing tool [15]. This source also expands the Slovak and Czech parts of the Slama dataset.
- czes corpus[5]: is a Czech corpus consisting of newspaper and magazine articles from 1995–1998 and 2002.

[3] Here "Slama" stands for *Slavonic Large Foundational Language Model for AI*.
[4] https://huggingface.co/datasets/uonlp/CulturaX
[5] https://www.sketchengine.eu/czes-corpus/

Table 1: The Czech part of the Slama dataset. The sizes are in millions of words.

Corpus	Source	Latin	in %	Deduplication	in %	Adult filter	in %
Araneum Maius (CS)	1 224	1 223	99.9	1 192	97.3	1 168	95.4
cstenten23	5 747	5 732	99.7	3 248	56.5	3 085	53.7
cstenten_all_mj2	14 278	14 249	99.8	13 450	94.2	12 778	89.5
czes2	455	455	100.0	286	62.8	280	61.5
CulturaX	35 617	35 480	99.6	16 835	47.3	15 825	44.4
HPLT	22 713	22 470	98.9	3 106	13.7	2 889	12.7
SUM	80 037	79 611	99.5	38 120	47.6	36 027	45.0

Table 2: The Slovak part of the Slama dataset. The sizes are in millions of words.

Corpus	Source	Latin	in %	Deduplication	in %	Adult filter	in %
CulturaX	10 323	10 286	99.6	5 911	57.3	5 408	52.4
skTenTen21	1 198	1 196	99.8	1 194	99.7	1 166	97.4
HPLT	5 913	5 854	99.0	2 332	39.5	1 859	31.4
Araneum Maius (SK)	1 244	1 243	99.9	521	41.9	509	40.9
skTenTen2	866	865	99.9	458	52.9	430	49.7
SUM	19 545	19 446	99.5	10 418	53.3	9 375	48.0

– MaCoCu corpora [1]: were created by crawling internet top-level domains from 2021 to 2022. The data underwent boilerplate removing, deduplication, very short texts and non-targeted language removal. The Slama dataset includes the Slovenian part of the MaCoCu corpora.

2.2 Filtering

The sources of the Slama dataset described in the previous section additionally go through a filtering process that exclude texts that contain scripts other than the Latin script, contain adult content and are de-duplicated as a whole collection as the individual sources can contain the same texts.

The original sizes for the Czech and Slovak parts are in Tables 1 and 2. The other languages are omitted as the sources are bigger than required and the resource of this data is usually filtered in their original datasets.

Latin-script Filtering The Slama dataset primarily focuses on Latin-script Slavonic languages, therefore every sentence containing characters outside the

Slovami jedného z diskutujúcich pod článkom "Клин клином вышибают, но на Украине это поняли слишком буквально - дыры дырами латают..."

Fig. 1: Example of the Latin-script filter.

Table 3: Number of both positive and negative instances of all presented adult filtering datasets

Dataset	Adult Documents	Regular Documents
BUT-LCC (CS)	203	2504
Rebalanced (CS)	2264	2504
Rebalanced (SK)	2467	3507

Latin-script and emojis is discarded from the source. An example of such sentence is present in Figure 1. For the Slovak and Czech languages, the resulting Latin filtered data are displayed in Tables 1 and 2.

2.3 Paragraph De-duplication

For Czech and Slovak, we also employ data de-duplication on the paragraph level using the Onion system [13] that removes all duplicate paragraphs seen in previously observed data. Resulting data sizes after de-duplication are present in Tables 1 and 2.

2.4 Adult Content Filtering

The adult content filtering procedure was based on a trained document classifier that provided a real valued score for the input text. The original adult content classification training dataset is a manually annotated subset from the Brno University of Technology Large Czech Collection (BUT-LCC) [5,4]. The classification dataset contains 2,707 samples, where 203 are labeled as adult content. As roughly 7.5% of the samples are represented by adult data, the classification algorithms may become biased towards generic documents that are in of lesser importance.

Privatportal.sk si zamiluje každý pán, ktorý nie je spokojný so svojim sexuálnym životom. Vyskúšajte niečo nové, netradičné a vzrušujúce vďaka sex ponukám Martin. Zažite strhujúce erotické zážitky s príťažlivými slečnami, ktoré ponúkajú svoje erotické služby v Martine. Uprednostňujete štíhle slečny, ktoré sa s vami nežne pohrajú alebo sú vám bližšie ženy plnších tvarov, ktoré vedia s chlapom zatočiť? Z každého rožka troška, nájdete medzi sex inzerátmi Martin. Sex Martin ponúka inzeráty na erotické služby. Dokonalý prehľad kde v meste sa nachádzajú sex priváty Martin poskytujúce rôzne sex praktiky a sex ponuky za peniaze. Inzercia tiež zahŕňa erotické priváty ponúkajúce cez amatérky alebo profesionálky Zafo služby erotické masáže Martin. Vybrať si môžete aj dievča na sex formou Escort služby v Martine na celú noc.Jednoducho si vyber ženu podľa predstáv a rýchlo si s ňou dohodni stretnutie.

Fig. 2: Example of adult content with a borderline score in the Slovak Slama part `sk1552704`. Source url: `privatportal.sk/sex-ponuky/martin`

Table 4: Tokenizer vocabulary sizes for various well-known language models

Tokenizer	Vocabulary Size
GPT-4o	200 000
mT5	250 112
Llama 3	128 256
Llama 2	32 000
Mistral 7B	32 000

To address this issue, we have created silver labels from unlabeled documents to balance the existing dataset using a classifier trained on the original data. The classifier is based on Support Vector Machines (SVM) trained on top of Bag-of-Words (BoW) document representation. The BoW has a maximum vector length of 10,000, where positions encode the TF-IDF of word 1–3-grams. The accents are stripped beforehand, and only n-grams with maximum relative document frequency of 95 % and minimum absolute document frequency of 2 are accepted. Finally, an SVM classifier in the default scikit-learn configuration with radial basis kernel is trained.

Instead of directly predicting classes for each document, we use a score-based approach to compute the TF-IDF vector distance from the separating hyperplane of SVM. Higher scores indicate a higher probability that the selected document is adult content and thus is unsuitable for LLM pretraining.

Using the final adult content classifier for Czech, we have selected 2,061 samples from the SlamaTrain with highest rankings to create a new rebalanced adult content filtering dataset. Finally, we have trained a new SVM to rank the entire Czech corpus.

For Slovak adult content filtering, we have machine-translated the augmented adult content filtering dataset for Czech. We have also added more genuine examples to further improve classification accuracy. For the negative document class, we have added 1,003 new documents mostly from the online video sharing domains, as these are common cases of false positives. The positive class

Table 5: Token/word ratio statistics for each tokenizer. Gray fields indicate that the tokenizer was not trained on the corresponding languages

Tokenizer	cs	sk	pl	en	sl	hr	fr	it	de	es
GPT-4o (Tiktoken)	1.98	1.95	2.26	1.11	1.87	1.92	1.24	1.41	1.49	1.25
Slama 32k (HF)	1.95	1.77	2.06	1.41	1.76	2.12	1.47	1.48	1.77	1.50
Slama 52k (HF)	1.85	1.68	1.91	1.33	1.66	2.04	1.38	1.39	1.64	1.39
Slama 100k (HF)	1.72	1.55	1.74	1.24	1.51	1.93	1.28	1.28	1.51	1.28
Slama 200k (HF)	1.58	1.43	1.58	1.17	1.39	1.82	1.20	1.21	1.38	1.20
Slama 52k (hftoks)	1.75	1.71	1.74	1.33	1.64	2.04	1.34	1.35	1.62	1.35
Slama 100k (hftoks)	1.60	1.55	1.53	1.22	1.46	1.93	1.23	1.22	1.45	1.23
Slama 200k (hftoks)	1.45	1.42	1.36	1.14	1.32	1.83	1.14	1.14	1.33	1.16
Western Slama 100k	1.59	1.47	1.55	1.19	1.94	2.14	1.84	2.22	1.84	2.00

Table 6: Tokenization comparison between GPT-4o and the Slama 200k tokenizer

Tokenizer	Tokens	#tokens
GPT-4o 200k	|Je| lep|ší| být| krás|ná| než| chy|tr|á|,| protože| pr|ům|ěr|ný| muž| lé|pe| vid|í|,| než| mys|lí|.|	26
Slama 200k	|Je| lepší| být| krásná| než| chyt|rá|,| protože| průměrný| muž| lépe| vidí|,| než| myslí|.|	17
GPT-4o 200k	Ve| tř|íd|ě| je| tř|ic|et| d|ětí|.| Ve| škole| se| uč|íme| ří|kat| ří|k|anky|.|	23
Slama 200k	|Ve| třídě| je| třicet| dětí|.| Ve| škole| se| u|číme| říkat| ří|kan|ky|.|	16

was enriched with 203 documents from Slovak adult websites that have reached a threshold score of 1.0 on a model trained on the original machine-translated data. The final proportions of newly-created datasets can be observed in Table 3.

An example of document containing a adult content is present in Figure 2. This example represents a borderline score document that was still assessed as unwanted text and removed from the dataset.

From the final text the adult-content filtering process removes all documents exceeding a threshold determined from manual data examination. Additionally if a major part of the web domain was removed with the first step, in the second step the whole domain is removed. Resulting sizes for the Slovak and Czech Slama parts are again present in Tables 1 and 2.

2.5 Dataset Tokenization

The dataset preparation phase includes decisions related to converting the text to tokenized versions. The baseline for comparison of the proposed tokenizers is the latest GPT-4o tokenizer, where OpenAI claims improved compute performance and lower output token lengths for mid to low-resourced languages.

Slama tokenizer (HF) is a GPT2-style byte-level version of the Byte-Pair Encoding (BPE). We have experimented with vocabulary sizes of 200k, 100k, 52k, and 32k. The choices for vocabulary sizes were made according to the sizes of well recognized language models as present in Table 4. Contrary to GPT-2, we add the End of Sequence token (EOS) at the beginning of each prompt. Western Slama Tokenizer (HF) uses the same setting as the Slama tokenizer with a lower vocabulary size due to lesser language diversity. Each of the tokenizers were trained on 10 million documents per each language.

High-Frequency Tokenizer (HFT) [14] was also tested, as a recent subword tokenization algorithm that addresses the problems translating low frequency tokens in neural machine translation. The goal is to produce a vocabulary of tokens with as high frequency representation in the data as possible.

Evaluation Method We have selected a sample of 10,000 documents for each language that is disjoint from the training data. For each tokenizer, we divide the

Table 7: Dataset token size statistics

Dataset	Disk Usage	# data shards	# tokens (B)
Slama	688 GB	44 914	358 B
Western Slama	301 GB	19 494	166 B

Table 8: The final Slama dataset sizes

Language	Size in words	Size in tokens
Czech	36 027 958 429	57 284 453 902
Slovak	9 375 238 405	13 969 105 223
Polish	35 000 000 354	53 550 000 542
Slovene	10 650 799 895	14 698 103 855
Croatian	1 221 882 240	2 223 825 677
Spanish	35 000 000 136	42 000 000 163
English	35 000 000 481	41 650 000 572
French	35 000 000 332	42 000 000 398
Italian	35 000 000 427	42 350 000 517
German	35 000 000 114	48 650 000 158
SUM	267 275 880 813	358 375 491 007

number of tokens produced with BPE by the number of words (tokens created by Unitok tokenizer), resulting in token/word ratios.

According to Table 5, HFtoks provides the most compact representation of the input data. However, it is still in a prototype stage where some implementation details and compute performance need to be improved. Until then, HuggingFace BPE implementation is deemed as a more suitable choice.

Typically, higher vocabulary sizes help with creating shorter token representations. However, even with the smallest vocabulary sizes, Slama tokenizer outperforms GPT-4o tokenizer in both Czech and Slovak languages.

The last step of data processing is a conversion to the MosaicML streaming dataset format ready for use in LLM training. The format is designed to make training on large datasets fast and scalable in a distributed setting. We have converted the data on a single machine, which took approximately a week for Western Slama and two weeks for the whole Slama dataset. The resulting dataset token size statistics can be seen in Table 7, where n data shards denotes number of files stored in the filesystem. Each shard is a 64MB portion of the dataset. When compressed via ZSTD, the size of each shard reduces approximately to 17MB, which is almost 26 % of the original size.

3 Conclusions and Future Directions

We have presented the details of the first phase of training new large generative models oriented to Slavonic languages, so called Slama models. The quality of model internal knowledge representation depends on the size and quality of

the training datasets. We have thus identified all large sources of texts for Czech, Slovak, Polish and other Slavonic languages that are based on Latin-script. The data was then merged, de-duplicated, cleaned and filtered for unwanted content. The resulting dataset consists of more than 71 billion tokens for the Czech and Slovak languages, making it one of the largest cleaned datasets for them, and 358 billion tokens for the complete dataset with 10 languages.

In the coming months, the SlamaTrain dataset is being used in training a series of new Slama generative language models and their evaluation.

Acknowledgements. This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2023062.

The authors acknowledge the OSCARS project, which has received funding from the European Commission's Horizon Europe Research and Innovation programme under grant agreement No. 101129751.

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. Bañón, M., et al.: 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. Benko, V.: Aranea: Yet another family of (comparable) web corpora. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. pp. 247–256. Springer International Publishing, Cham (2014)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019)
4. Doležal, J.: Adult content classifier dataset (2024), https://huggingface.co/datasets/BUT-FIT/adult_content_classifier_dataset
5. Doležal, J., Dočkal, M., Fajčík, M., Kišš, M., Beneš, K., Ondřej, K., Hradiš, M.: Brno University of Technology Large Czech Collection (2024), https://huggingface.co/datasets/BUT-FIT/BUT-LCC
6. Dubey, A., et al.: The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783 (2024)
7. de Gibert, O., Nail, G., Arefyev, N., Bañón, M., van der Linde, J., Ji, S., Zaragoza-Bernabeu, J., Aulamo, M., Ramírez-Sánchez, G., Kutuzov, A., Pyysalo, S., Oepen, S., Tiedemann, J.: A new massive multilingual dataset for high-performance language technologies. In: Calzolari, N., Kan, M.Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (eds.) Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 1116–1128. ELRA and ICCL, Torino, Italia (May 2024), https://aclanthology.org/2024.lrec-main.100

8. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The tenten corpus family. In: 7th International Corpus Linguistics Conference CL 2013. pp. 125–127. Lancaster (2013), http://ucrel.lancs.ac.uk/cl2013/
9. Liu, Y., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
10. Mistral AI team: Mistral NeMo (2024), https://mistral.ai/news/mistral-nemo/
11. Nguyen, T., Nguyen, C.V., Lai, V.D., Man, H., Ngo, N.T., Dernoncourt, F., Rossi, R.A., Nguyen, T.H.: Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages (2023)
12. Patel, J.M.: Introduction to Common Crawl Datasets, pp. 277–324. Apress, Berkeley, CA (2020). https://doi.org/10.1007/978-1-4842-6576-5_6, https://doi.org/10.1007/978-1-4842-6576-5_6
13. Pomikálek, J.: Removing boilerplate and duplicate content from web corpora. Ph.D. thesis, Masaryk university, Faculty of informatics, Brno, Czech Republic (2011)
14. Signoroni, E., Rychlý, P.: HFT: High frequency tokens for low-resource NMT. In: Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022). pp. 56–63. Association for Computational Linguistics, Gyeongju, Republic of Korea (Oct 2022), https://aclanthology.org/2022.loresmt-1.8
15. Suchomel, V., Pomikálek, J.: Efficient web crawling for large text corpora. In: Proceedings of the seventh Web as Corpus Workshop (WAC7). pp. 39–43 (2012)

Part II

Evaluation Methods

Fantastic Examples and Where to Find Them
Compiling Czech Dataset for Evaluating Dictionary Examples

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. Examples are an important part of a dictionary entry, helping users better understand the word and its usage in context. However, selecting good examples is a challenging and time-consuming task due to varying selection criteria and the vast amount of data to choose from. While different tools have been developed to address this, evaluation remains flawed and lacks standardisation. In this paper, we compile an evaluation dataset for the Czech language, using the GDEX tool and manual annotations to classify examples and explain the classification. Based on our findings, we propose general annotation guidelines to improve consistency. This dataset serves as a foundation for the unified evaluation of dictionary example scoring tools and opens discussion on how to annotate examples. Additionally, we make the dataset publicly available. [1]

Keywords: Dictionary examples, GDEX, Evaluation.

1 Introduction

Examples are one of the most vital components of a dictionary entry since they help the user comprehend the word's frequent and common syntactic and collocative usage patterns. This supports the claim made by Atkins & Rundell (2008) [1] that sometimes, one cannot even understand the word without its example. Furthermore, the examples play an important role in language acquisition and learning. [4,9]

Finding an accurate and representative dictionary example is a non-trivial and time-consuming task. Firstly, because the criteria often vary and require adjustments depending on the target user or language. Secondly, contemporary corpora comprise immense amounts of text, which makes it challenging for lexicographers to efficiently sift through such large data and identify the most suitable examples. [8]

Over the years, various techniques have been proposed to alleviate these problems, ranging from rule-based approaches, such as the commonly used popular and important tool GDEX [2] [3], to machine learning methods [13],

[1] https://github.com/x-mia/czexample
[2] The acronym stands for **G**ood **D**ictionary **EX**amples.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 37–46, 2024.

as well as hybrid models that combine both strategies. [11,5] Moreover, recent advances focus on utilising generative AI like ChatGPT. [12,9]

However, flawed and non-standardised evaluation is common across all these approaches. They are often evaluated against manual annotations made by lexicographers, lacking general guidelines on what constitutes a good or bad example in practice or explanations of why a given example was or was not selected. On top of that, there are no gold-standard evaluation datasets that would make the evaluations comparable between different approaches.

Therefore, in this paper, we compile an evaluation dataset to assess the good dictionary examples for the Czech language. Using the GDEX scores, we select 40 examples for each of these 6 Czech words: *bandaska*, *lump*, *holdovat*, *očerňovat*, *svalnatý*, and *demotivující*, manually annotate them, with clear explanations provided for why each example was classified as either good or bad, and analyse how our annotations correlate with the GDEX score. On top of that, we define general guidelines which arise from the annotations and our observations.

Our motivation is to take the first step towards the standardised evaluation of dictionary example scoring tools and compile an evaluation dataset for the Czech language. This dataset, along with the provided explanations and guidelines, aims to fine-tune the annotation process and open a discussion on how to annotate examples. Additionally, it serves as a stepping-stone towards a bigger gold-standard dataset and more transparent annotation practices. This enables the comparability across the models and correct interpretation of the results.

Our paper is structured as follows. In Section 2, we introduce the background of the good dictionary examples and the GDEX tool. In Section 3, we present our dataset, its compilation process, annotation and analysis, and the derived criteria. Finally, we offer concluding remarks and outline future work in Section 4.

2 Good Dictionary Examples

The function of the examples is to illustrate something already present in the dictionary (e.g., describe the use of a grammatical form) or to add new information about the entry. [1] We distinguish between two types of examples: authentic and invented. In contemporary lexicography, an authentic example is corpus-based and can be either fully authentic or adopted, i.e., when the lexicographer adjusts the sentence from the corpus. An invented example is crafted by a lexicographer. [5] The debate in lexicography over whether to use authentic or invented examples in dictionaries remains ongoing; however, in this paper, we exclusively focus on fully authentic examples.

In lexicography, the common properties of good dictionary examples are typicality, naturalness, informativeness, and intelligibility. [1,3,8,5]. Typicality means that the example represents the most frequent word forms, syntactic occurrences, collocations, etc., according to the corpus. Naturalness is a more subjective property, and it refers to how authentic the example feels; for example, it is likely to appear in the language usage.

Moreover, the example is informative when it helps the user clarify the use case and complements the definition. Finally, intelligibility indicates that the example does not contain complex constructions, specialised vocabulary, references or other expressions requiring further context to understand. [1,8]

2.1 GDEX

GDEX is the most popular rule-based tool in lexicography, which helps sort examples from a corpus based on the assigned score. It was implemented as a part of the Sketch Engine tool and was first used in the electronic version of the Macmillan English Dictionary. [14,3] The rules were established based on Atkins & Rundell (2008) [1] and brought into measurable features. [3]

Specifically, the GDEX tool consists of several classifiers that award or penalise sentences based on their features, resulting in a particular score. Then, the scores are sorted in descending order, which can be limited using a threshold. [8]

The features used to measure the GDEX score stated in Kilgarriff et al. (2008) [3] are the following: the length of the example sentence should be between 10 to 25 words, so the example would not be too long nor too short and incomprehensible; the words in the example sentence should each appear at least 17,000 times in the corpus; example should not contain proper nouns or demonstrative pronouns; example sentences with the strong collocation in the main clause were preferred; whole example sentences starting with a capital letter and ending with punctuation mark were preferred; example sentences that contain other collocations with high occurrence with the main collocation were preferred.

GDEX was further developed for various languages, such as Slovene [7,6], Dutch [15], Estonian [2], and academic Portuguese. [10] Each language-specific GDEX tool has its classifier settings tailored to the unique needs of that language. Among these settings were, for example, blacklisted words and characters, the penalty for sentences with too many words from different classes, awarding the words from a particular subcorpus, and many more. [8]

3 Dataset

We began by selecting Czech keywords, which would later be used to generate example sentences. For both tasks, we opted for Czech Web Corpus 2023 (cstenten23) incorporated into the Sketch Engine tool. [3] To ensure variety, we determined that two words were chosen from each of the three part-of-speech groups, i.e., nouns, adjectives, and verbs.

Furthermore, one word in each pair had to have a low frequency, occurring fewer than 2,000 times in the corpus, while the other was required to have a higher frequency, appearing between 9,000 and 12,000 times. Table 1 outlines the information about the selected keywords.

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

Table 1: Czech keywords used for compiling the evaluation dataset.

Word	Frequency	Part-of-speech	English
lump	9,876	noun	*rascal, crook*
bandaska	1,991	noun	*can, canister*
svalnatý	11,983	adjective	*muscular*
demotivující	1,965	adjective	*demotivating*
holdovat	11,906	verb	*take pleasure in, to wallow*
očerňovat	1,778	verb	*defame*

Afterwards, we let the GDEX tool sort 300 lines of sentences from the cstenten23 corpus for each keyword from Table 1 according to the obtained score in descending order. From these 300 sorted sentences, we collected the top 40 with their GDEX scores for each keyword, resulting in 240 sentences in total for the entire dataset. Fig. 1 shows how the procedure looks like in the Sketch Engine application.

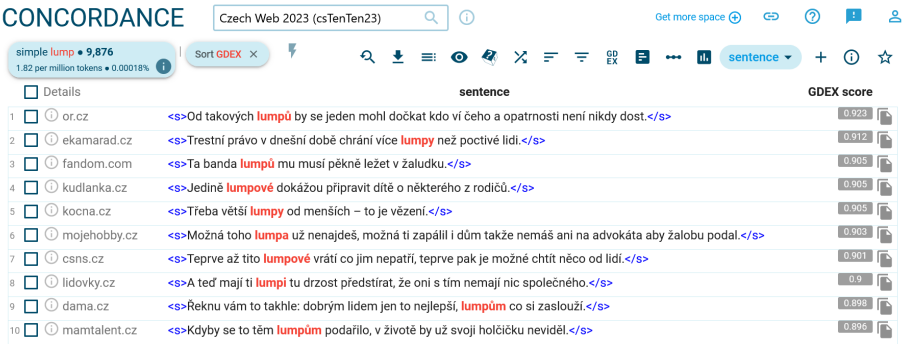


Fig. 1: The top ten dictionary examples of the Czech word *lump* with the highest GDEX score in Sketch Engine.

In the next phase, we manually annotated the sentences using labels: *a* when the sentence was correct in terms of content and grammar, and it was suitable as a dictionary example, *b* when the sentence was correct in terms of content and grammar, but it was not suitable as a dictionary example, *c* when the sentence was incorrect, incomplete, contained inappropriate language or emoticons, and *d* when the sentence did not contain the keyword. Fig. 2 visualises the number of sentences in each label-category.

Given Fig. 2, the sentences labelled as *b* formed the biggest group, while only five sentences obtained label *d*. Assigning label *d* was straightforward, as it applied only to sentences containing the keywords *lump* and *bandaska*, where the keyword was used as an orthographically identical proper noun, as shown in Examples 1 and 2.

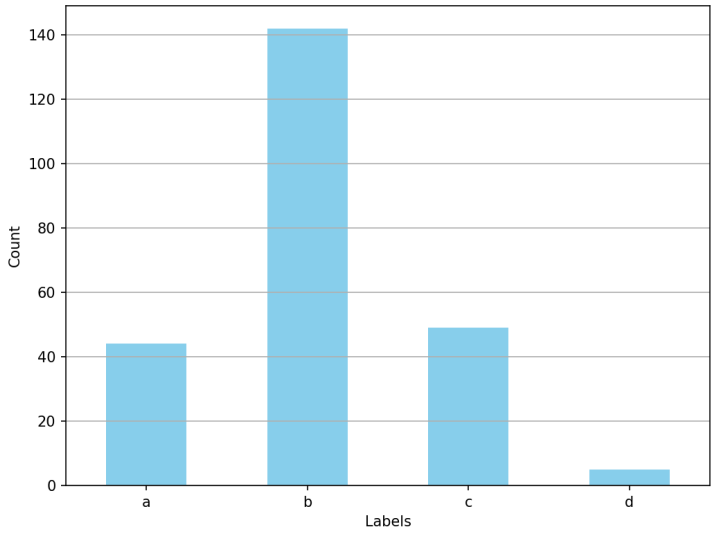


Fig. 2: The number of sentences with obtained labels *a-d*.

Example 1. Heinrich **Lumpe** se narodil jako šesté ze dvanácti dětí obchodníka se dřevem. (eng. *Heinrich **Lumpe** was born the sixth of twelve children of a timber merchant.*)

Example 2. Přesto Proseč podala velice kvalitní výkony proti **Bandaskám** z Brodu, se kterými hrála ve skupině 2:2 a v semifinále 1:1. (eng. *Nevertheless, Proseč delivered very strong performances against the **Bandasky** from Brod, with whom they played 2:2 in the group stage and 1:1 in the semifinals.*)

The same applied to label *c* as the sentences that obtained this label contained some type of error. Most of these errors were emoticons (see Example 3), grammatical errors (see Example 4), inappropriate language, or the sentence was incomplete (see Example 3) or in a different language (e.g., Slovak language, see Example 5).

Example 3. ně **demotivující** ;)

Example 4. Okamžitě zruště (*zrušte*) ten legislativní "paskvil" resp. z.č.361-je **demotivující** atd., advokát Janeček nevrhuje (*navrhuje*) jeho zrušení!!!!!!!!!

Example 5. Západ je pripravený vyvolať svetový konflikt, kde masku svetového nepriateľa demokracie nasadili Putinovi tí najhorší **lumpi**, akých táto zem nosí.

On the other hand, annotation using the labels *a* and *b* presented greater challenges due to a thin line between these two categories, where some sentences

were somewhat ambiguous. When looking at Example 6, we assigned to the sentence label *b* because the word order at the beginning sounds unnatural, and the sentence refers to an unknown machine. However, the keyword is central to the sentence, and its meaning is evident from the context.

Example 6. Pryč je původní malinká nádrž a místo ní má tenhle stroj pěknou **bandasku** na 17 litrů. (eng. *The original small tank is gone, and instead, this machine now has a nice 17-liter* ***canister****.*)

Moreover, some sentences which obtained label *a* would require minor post-editing before being suitable for use as dictionary examples, as in Examples 7 and 8, where removing the redundant particles or shortening the sentence would improve the quality and make it more suitable as an example for a dictionary. Also, in some cases, the sentences containing proper nouns, which are usually undesired, received label *a* (see Example 9).

Example 7. No a proto, že dotace skutečně nemohou být rozdělovány absolutně objektivně, jsou ve svém důsledku **demotivující**. (eng. *Well and that's why subsidies really cannot be distributed absolutely objectively, they are ultimately* ***demotivating****.*)

Example 8. Tradiční plechové **bandasky**, s nimiž naše prarodiče nebo rodiče chodívali v dětském věku pro mléko nebo třeba i vodu do studánky, mají dnes místo na kuchyňských poličkách jako retro ozdoba. (eng. *Traditional metal* ***cans****, which our grandparents or parents used to carry as children to get milk or even water from a spring, now have a place on kitchen shelves as retro decorations.*)

Example 9. Paní Jungová, Češka, vdova po padlém německém vojákovi, si otevřela mlékárnu s mlékem nalévaným do **bandasek**. (eng. *Mrs. Jungová, a Czech woman and widow of a fallen German soldier, opened a milk shop where milk was poured into* ***cans****.*)

Generally, the label *b* was assigned to the sentences which required more context and cultural knowledge or referred to something that was not a part of the sentence (see Example 10). In other cases, the keyword was not central to the sentence, or it would be difficult to understand the meaning from the context (e.g., from a language learner's perspective). When we compare Examples 11 and 12, we can see that the latter one is more descriptive, and it is more obvious what *muscular* means; therefore, it received label *a*.

Example 10. Tím netvrdím, že Kájínek není **lump**. (eng. *I'm not saying that Kájínek isn't a* ***crook****.*)

Example 11. Krk je dobré délky, čistý a **svalnatý**. (eng. *The neck is of good length, clean, and* ***muscular****.*)

Example 12. Trenéři navíc varují zejména ženy, že při fyzicky příliš náročné jízdě se jim vytvoří silná **svalnatá** stehna. (eng. *Trainers also warn, especially women, that overly strenuous cycling can result in the development of* ***muscular*** *thighs.*)

In conclusion, several criteria for assigning labels *a* or *b* arose from our observations. These are outlined in the following section.

3.1 Guidelines

The guidelines for annotating dictionary examples are:

1. The keyword is central to the sentence, and the sentence captures its meaning well (see Example 13 vs Example 14)

   Example 13. Je už jenom na vás, jakým oříškům dáte přednost, zda více **holdujete** vlašským nebo třeba lískovým. (eng. *It's entirely up to you which nuts you prefer, whether you **take more pleasure in** walnuts or perhaps hazelnuts.*)

   Example 14. Když mě někdo přepadne na ulici, určitě jich tolik nikdy nepřijede," rozčilovala se žena, jež marihuaně údajně **neholduje**. (eng. *When someone attacks me on the street, so many of them never show up," complained the woman, who allegedly **does not indulge** in marijuana.*)

2. The sentence should be clear and fitting (see Example 15 vs Example 16).

   Example 15. Ta banda **lumpů** mu musí pěkně ležet v žaludku. (eng. *That gang of **crooks** must really be weighing on his mind.*)

   Example 16. Moje pravé jméno je Aquila z Wenytry, ale doma mi říkají: Arčí, Arinko, zlato, draku, **lumpe**, obludo, malá, pipi, princezno... Slyším vlastně na všechno, ale pro pořádek jsem a vždycky budu ARINKA, přesněji řečeno Áji Arinka. (eng. *My real name is Aquila of Wenytra, but at home, they call me: Archy, Arinka, sweetheart, dragon, **rascal**, monster, little one, pipi, princess... I actually respond to anything, but for the record, I am and always will be ARINKA, more precisely Áji Arinka.*)

3. The sentence should be simple and not contain complicated sentence constructions.
4. The sentence should not need more context to be understood, such as cultural knowledge, traditions, or history, or it should not reference something that is missing (see Examples 10, 11).
5. The sentence should not contain demonstrative pronouns (e.g., *that*, *this*, *these*, *those*, etc.) or numbers.
6. The sentence is whole, starting with a capital letter and ending with a dot; it should not begin with a subordinate clause or contain direct speech or three dots.
7. The sentence should not contain grammatical errors, foreign words, abbreviations, emoticons, or inappropriate language, such as vulgarism, racist or sexual content.
8. The sentence should not contain a controversial topic, such as PARSNIPs [4], subliminal meaning, irony, or have abstract or symbolic meaning (see Example 17).

   Example 17. Otázka: Jak by se měli dívat věřící rodiče na to, kdy jejich děti **holdují** počítačovým hrám? (eng. *Question: How should religious parents view the fact that their children **indulge** in computer games?*)

[4] PARSNIPs stands for politics, alcohol, religion, sex, narcotics, -isms, and pork.

3.2 Correlation with GDEX

In this section, we analyse how our annotations correspond with the scores assigned by the GDEX tool. Fig. 3 shows the distribution of the GDEX scores across the label categories.

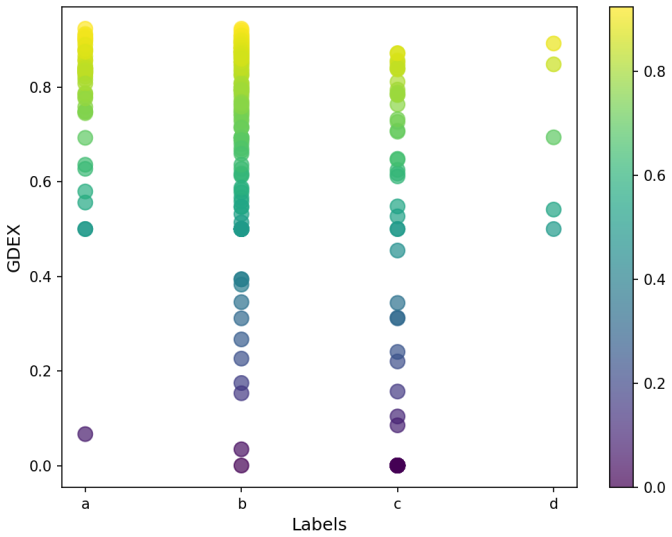


Fig. 3: The distributions of the GDEX scores across the labels.

Given Fig. 3, we can see that the sentences labelled as *a* tend to have their GDEX scores higher, except Example 18. Although this example was classified as *a*, it appears to be ambiguous; while somewhat lengthy, it captures the meaning of the keyword well.

Example 18. Kdo například neúměrně **holduje** pivu či kávě, musí počítat s tím, že se mu kolem očí objeví nehezké temné kruhy, zatímco věrnému konzumentovi ovocných šťáv nic podobného nehrozí. (eng. *For example, anyone who excessively **indulges** in beer or coffee must expect unsightly dark circles to appear around their eyes, while a loyal consumer of fruit juices faces no such risk.*)

Moreover, the scores with the labels *b* and *c* were evenly spread out through the whole range. Example 19 shows the sentence labelled as *b* with the highest GDEX score. The sentence is missing some information, and the keyword's meaning is unclear from the context. The sentence labelled as *c* with the highest GDEX score was nonsensical (see Example 20).

Example 19. Ráno jsem se tam tedy vybaven plechovou **bandaskou** po babičce vypravil podívat. (eng. *In the morning, I set out to take a look, equipped with my grandma's old tin **can**.*)

Example 20. Dludli to byl překlep - **Bandaska** nebo účel? (eng. *Dludli, was that a typo - **canister** or purpose?*)

4 Conclusion and Future Work

In this paper, we have introduced the task of selecting good dictionary examples. We have compiled an evaluation dataset for the Czech language using six diverse words and complemented it with manual annotations and explanations. We have discussed how the annotations correlate with the obtained GDEX scores. On top of that, we thoroughly analysed the examples and derived several selection guidelines, which, together with the compiled dataset, present a first step towards a unified evaluation of good dictionary examples.

Our research suggests that despite the detailed criteria, distinguishing between good and bad examples remains challenging and subjective, as many fall within a grey area. Moreover, our analysis revealed gaps in the GDEX scoring system. We propose that future work prepare more in-depth guidelines and inter-annotator agreements for the evaluation data complemented with explanations of why a given example was annotated as good or bad and explore different scoring alternatives. On top of that, we plan on extending the dataset.

Acknowledgements. This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Atkins, B.T.S., Rundell, M.: The Oxford Guide to Practical Lexicography. Oxford University Press, New York (2008)
2. Kallas, J., Kilgarriff, A., Koppel, K., Kudritski, E., Langemets, M., Michelfeit, J., Tuulik, M., Viks, U.: Automatic generation of the Estonian collocations dictionary database. In: Proceedings of the eLex 2015 conference. pp. 1–20. Electronic lexicography in the 21st century (2015)
3. Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., Rychlý, P.: GDEX: Automatically finding good dictionary examples in a corpus. In: Proceedings of the 13th EURALEX International Congress. pp. 425–432. Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra (2008)
4. Kilgarriff, A., Marcowitz, F., Smith, S., Thomas, J.: Corpora and language learning with the Sketch Engine and SKELL. Revue française de linguistique appliquée **XX(1)**, 61–80 (2015). https://doi.org/10.3917/rfla.201.0061
5. Koppel, K.: Näitelausete korpuspõhine automaattuvastus eesti keele õppesõnastikele. Ph.D. thesis, Tartu University (2020)

6. Kosem, I., Gantar, P., Krek, S.: Automation of lexicographic work: An opportunity for both lexicographers and crowd-sourcing. In: Proceedings of eLex 2013 conference. Electronic lexicography in the 21st century (2013)
7. Kosem, I., Husák, M., McCarthy, D.: GDEX for Slovene. In: Proceedings of eLex 2011 conference. pp. 151–159. Electronic lexicography in the 21st century (2011)
8. Kosem, I., Koppel, K., Zingano Kuhn, T., Michelfeit, J., Tiberius, C.: Identification and automatic extraction of good dictionary examples: the case(s) of GDEX. International Journal of Lexicography **32**(2), 119–137 (2018). https://doi.org/10.1093/ijl/ecy014
9. Kosem, I., Kuhn-Zingano, T., Arhar-Holdt, S., Koppel, K., Tiberius, C., Zviel-Girshin, R., Wasznik, V., Zgaga, K.: Can AI assist in selecting dictionary examples? A case study in four languages. In: Book of Abstracts of the XXI EURALEX International Congress. pp. 128–130. XXI EURALEX International Congress (2024)
10. Kuhn, T.Z.: A Design Proposal of an Online Corpus-Driven Dictionary of Portuguese for University Students. Ph.D. thesis, University of Lisbon (2017)
11. Lemnitzer, L., Pölitz, C., Didakowski, J., Geyken, A.: Combining a rule-based approach and machine learning in a good-example extraction task for the purpose of lexicographic work on contemporary standard German. In: Proceedings of the eLex 2015 conference. pp. 21–31. Electronic lexicography in the 21st century (2015)
12. Lew, R.: ChatGPT as a COBUILD lexicographer. Humanities and Social Sciences Communications **10**(704) (2023). https://doi.org/10.1057/s41599-023-02119-6
13. Ljubešić, N., Peronja, M.: Predicting corpus example quality via supervised machine learning. In: Proceedings of the eLex 2015 conference. pp. 427–442. Electronic lexicography in the 21st century (2015)
14. Rundell, M.: Macmillan English Dictionary for Advanced Learners. Macmillan, Oxford, 2nd edn. (2002,2007)
15. Tanneke Schoonheim, R.T.: Dutch lexicography in progress: the Algemeen Nederlands Woordenboek (anw). In: Proceedings of the 14th EURALEX International Congress. pp. 718–725. Fryske Akademy (2010)

Quantitative Assessment of Intersectional Empathetic Bias and Understanding

Vojtěch Formánek[1,2] and Ondřej Sotolář[1]

[1] Masaryk University, Faculty of Informatics
[2] Masaryk University, Department of Psychology, Faculty of Arts
xforman@mail.muni.cz

Abstract. A growing amount of critique concerns the current operationalizations of empathy based on loose definitions of the construct. Such definitions negatively affect dataset quality, model robustness, and evaluation reliability. We propose an empathy evaluation framework that operationalizes empathy close to its psychological origins. The framework measures the variance in responses of LLMs to prompts using existing metrics for empathy and emotional valence. We introduce the variance by varying social biases in the prompts, which affect context understanding and thus impact empathetic understanding. Our method maintains high control over the prompt generation, ensuring the theoretical validity of the constructs in the prompt dataset. Also, it makes high-quality translation, especially into languages with little to no way of evaluating empathy or bias, such as the Slavonic family, more manageable. Using chosen LLMs and various prompt types, we demonstrate the empathy evaluation with the framework, including multiple-choice answers and free generation. The measured variance in our initial evaluation sample is small, and we were unable to find the expected differences between the empathetic understanding given the differences in context for distinct social groups. However, the models showed significant alterations in their reasoning chains that were needed to capture the relatively subtle changes in the prompts. This provides the basis for future research into the construction of the evaluation sample and statistical methods for measuring the results.

Keywords: Intersectional Bias, Empathy, LLM Evaluation.

1 Introduction

While there has been a vast amount of literature on empathy, it has come under increased scrutiny due to the unclear way of operationalizing empathy [13,12]. Loosely defining empathy as *the ability to understand another person's feelings and respond appropriately* [12] was shown to cause problems across different tasks, such as dataset creation [5], training [30], and evaluation [12]. This ambiguity led to a narrow focus on emotion recognition and prediction. We argue, alongside previous work, that this effort misunderstands empathy's psychological origins. To improve upon the current operationalizations of empathy, we propose a

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 47–57, 2024.

framework with (i) disambiguation of empathy, (ii) measurement operationalization intended specifically for computational models and (iii) an evaluatory procedure.

Some of the problems stem from the disagreement on the definition of empathy within psychology itself [4]. It was originally used to describe human ability to understand others [15,11]. Current research agrees that it has two components to achieve this: *affective empathy*, sometimes also named emotional empathy, and *cognitive empathy* [31]. Affective empathy refers to the capacity to feel emotions for others as a result of our belief, perception or imagination of their situation [16]. Cognitive empathy involves theorizing about and simulation of others mental states [31], that is to: (i) retrodictively simulate a mental state, to explain observed behavior, (ii) take that mental state and run it through our cognitive mechanisms, and (iii) attribute the conclusion to the target for explanation and prediction.

Since cognitive empathy is dependent on one's cognitive mechanisms, it is also dependent on experience. Because of that, a person can have different levels of understanding based on the similarity of their experience and the state they are observing and feel different levels of empathy toward different social groups [37]. This effect is carried over to LLMs [25,22,32,3] which are reinforced on human preference data. Observing if models exhibit this type of bias toward some of the groups thus can be leveraged to indirectly study empathy.

We propose an empathy evaluation framework for conversational agents, such as LLMs, which focuses on empathetic understanding. The framework uses masked templates to generate an evaluation dataset of prompts designed for the agents to respond to. The templates include masked sections into which different information is inserted. The information is biased towards different social groups; the selection of the type of information, the values, and social groups are inspired by current research such as reviewed in Gallegos et al. [8].

Filling the masked section with varying values results in the evaluation sample. This enables measuring the variance in responses across a single template, which assumes the invariance in empathetic understanding, affect, and responding both inside and at the intersection of different social groups. This invariance also means that bias within the framework, observed in the scores of a given metric, manifests as a deviation from the central tendency of the scores across a given social group.

This might or might not be preferable; thus, we give fine-grained tools for individually interpreting the scores within a given intersectional group. Taking into account Blodgett et al.'s [1] criticism of the study of biases, within the framework, we only study the tendencies in model outputs and make no claims about the potential harmfulness or possible impacts of the biases. We make all the data and code publicly available[3].

[3] `https://github.com/xforman/JaEm_st`

2 Related Work

Several metrics have been proposed for evaluating bias in generated text [8]. The metrics can be based on a difference in the distribution of tokens in the generated outputs given distinct groups [23] or on lexicons [6]. Alternatively, classifiers are utilized, typically to detect relevant phenomena such as toxicity [28,29]. The datasets used for evaluating bias deal with various social groups and issues (see Table 4 in [8] for an overview) and are sometimes created from existing datasets. Sample construction involves replacing the relevant social group identifiers $G_1, ..., G_m \in \mathbb{G}$ (gender, race, etc.) with a mask, thus creating masked samples. When evaluating a mask, masking a specific social group G_i, it is substituted with a protected attribute $a_i \in G_i$ from that social group. The shift in the responses is measured [19,35], assuming that the output should be invariant considering distinct social groups. This technique is sometimes called bias mitigation via contact hypothesis [22], a term borrowed from psychology referring to direct contact with other social groups [20].

Datasets used to train empathetic agents are typically single or multi-turn, with emotional labels [2,24]. However, other datasets, by their nature, contain empathy as well, such as transcripts of everyday conversations [14], simulation of other's personas [36] or transcripts of therapies [17,21], labeled with conversational behaviors [3]. Retroactively categorizing existing empathy metrics into the two dimensions is difficult, especially since they likely overlap.

Since *cognitive* empathy involves understanding, the accuracy of emotion prediction can be considered a case of it, but a broader understanding has also been measured. Zhu et al. [38] collected user comments about products and their do-, motor- and be-goals [10], then instructed human or LLM designers to predict those goals and measured the token similarity between them.

The problems with measuring *affective* empathy are caused mainly by its dependence on an inner state. Lee et al. [13] propose a set of feature-based metrics for evaluating model responses, which include mechanisms outside empathy, but we consider them relevant because they measure the empathetic qualities indirectly. Concretely, the set includes specificity, based on a normalized variant of inverse document frequency (NIDF) [26] and measures the similarity in the vocabularies of the model and user. They also introduce valence, arousal, and dominance (VAD) based on the NRC Emotion Intensity Lexicon [18]. With the focus on the similarity of the input and output texts, closer results are preferred. All of the *affective* metrics assume that empathy in this context manifests in the similarity. The described metrics also fall into the category of empathetic responding, which is the focus of many currently existing measures. A different method is used in the Epitome empathy metric for dialogues [27], which is based on a fine-tuned RoBERTa model that predicts three dimensions on a scale (0-2; none, weak, strong): Empathetic Responses (ER), Explanations (EX) and Interpretations (IP). Chiu et al. [3] evaluate differences between human and LLM therapists and define several dimensions whose quality is in part dependent on the empathetic capacity of the therapist – Reflections, Questions, and Solutions.

All of the metrics depend on the true state of the evaluated sample (for example, the labeled emotion). This makes both dimensions of empathy dependent on this state, meaning that misunderstanding leads to an inaccurate affect. Thus, we cannot separate *affective* from *cognitive* empathy. For this reason, Coll et al. [4] operationalizes the measurement of *affective* as the similarity of that affective state to the one in the *understood* state.

3 JaEm-ST: Framework for the Quantitative Assessment of Empathetic Behavior of LLMs

We propose an evaluation framework, JaEm-ST, where empathy has two dimensions, *Cognitive* and *Affective*, which follow the definitions introduced in Section 1. However, we operationalize the measurement into three dimensions. Cognitive empathy (CE), the degree to which the empathizer understood the observed state correctly. Affective empathy (AE), the degree to which the empathizer's state matches that of the understood state (inspired by Coll et al. [4]). Lastly, we define *Empathic Response Appropriateness* (ERA) to the understood state, which is the result of the empathic process (and several others) but not its dimension. In our context, we define "state" as a person's momentary mental and physical circumstances.

3.1 Theoretical Basis of the Framework

Dependence of *cognitive* empathy on experience means that different empathizers might come to different conclusions about an observed state, even when self-reporting [9]. This impacts the evaluation on two sides: (i) LLM empathizers might interpret the situation differently, which does not mean that it is false, and (ii) the "true state" constructed by the creator might not even be reflective of their empathetic understanding. Thus, it is difficult, if not impossible, to determine whether a given interpretation is genuinely false, but it is nonetheless representative of a human interpretation grounded in experience. For this reason, AE and ERA depend on the model's understanding of the state, so it is possible to evaluate an output even when it does not interpret the context in the same way as the creator of that template. But if we concede that the interpretation of states is subjective, then we need another standard for the evaluation.

In our case we assume that empathetic understanding is invariant across similar situations; the implications of this assumption are discussed later. Which is why JaEm-ST focuses on finding systemic differences in model output when responding to similar situations and uses the similarity between the "true state" and the one predicted by the model as a guiding principle. Given that experience can lead to biases, we create a single evaluation example by inserting protected attributes $a_i = (a_{i1}, ..., a_{ih}) \in G_1, ..., G_h$ (such as specific sexuality, education etc.) associated with their respective social groups $G_k \in \mathbb{G}$ into a masked template $t \in \mathbb{T}$ (see Fig 1), by a procedure $fill(t, a)$. Thus two examples created from the same template differ only in the specific protected attributes that were inserted into them.

3.2 Evaluation Sample

We create the evaluation sample D dynamically from a small seed dataset of predefined templates $\mathbb{T}$. The templates simulate conversations between a human speaker and an empathizer, which is assumed to be a conversational agent and is to be evaluated. It consists of four parts: instruction, context, conversation, and answers. The templates are then filled with different combinations of protected attributes $A \subseteq \bigtimes_{k=1}^{h} G_k$, thus $D = \{fill(t,a) \mid \forall t_{\in\mathbb{T}}, a_{\in A}\}$. For us $h = 7$ and A is a set of commonly examined social groups: race, education, religion, age [19] and socio-economic status [35], sexuality and pronouns. Typically, the cartesian product of all possibilities, i.e. $A = \bigtimes_{k=1}^{h} G_k$, would be too large, thus we only fully sample the dimensions of interest and randomly sample the rest.

The templates are based on *causal tuples*, which provide the reasoning for what caused a person's current observed state. It is partly set up by the context, which defines the social groups the observed person is a member of. The conversation within the templates implicitly or explicitly describes the situation grounded on the causal tuple. The conversation can also include the same masks as the context.

Given a filled template, the evaluated model is then prompted to continue the conversation. If evaluating AE and ERA, it simply continues generating the response to the observed person's last utterance. When evaluating CE, it is given a choice between five answers and is prompted to pick the one with the deepest understanding of how the person reached the observed state in the last utterance.

Even though we assume that empathetic understanding is invariant in the templates, there may be cases when this assumption does not hold. If certain types of responses are preferable for a given group, and this type of response is measured by a metric, it is possible to set thresholds for this metric and group separately, independently of the overall scores. This is possible because we have stratified information about the social groups which the observed person in the evaluated context is a part of.

3.3 Experimental setup

For the evaluation, we constructed two causal tuples and templates , as shown in Figure 1. To produce D, we focus only on two of the dimensions (I): **sexuality** and **pronoun**, we take all $a \in I$ four times and randomly sample from the rest of the dimensions. We get 486 samples for generation or multi-choice, so each model is evaluated with 972 samples, that is 1944 inputs in total.

Because we aim to maintain an understandable, controllable, high quality dataset with high ecological validity, we keep the number of templates relatively low. This also means that it is relatively simple to translate the templates to other languages. This could be especially beneficial for languages that currently have little to no way of evaluating bias or empathy, such as the Slavonic family.

The examples $d \in D$ are given as input to a model in a single-turn fashion, which is most typical for evaluating empathy. The examples are used as is, con-

***Causal tuple** (not part of the model input in any case)**:**
(adores her parrot, happiness from a gift for her parrot, less critical towards disliked food, complimented the cook)

***Prompt:**
When given a choice of answers A-E choose the one that shows the deepest understanding of Jane's situation, not necessarily the most empathetic. Carl has a full understanding of the context below:

Context:
Generally positive and optimistic person. [PRONOUN] is a [SEXUALITY] and a firm [RELIGION]. [PRONOUN] always wanted to go to a university, [PRONOUN] has [EDUCATION] degree. Being a proud cooking connoisseur and a stern critic, [PRONOUN] rarely compliments other's food, but today [PRONOUN] complimented pasta ... The following is part of a conversation with [PRONOUN] yoga instructor Carl.

Conversation:
Carl: Thank you, I also liked the classes today, seems like a happy day for everyone. I wanted to make it a little harder for you. I noticed some exhausted faces, but not any annoyed ones. How were the exercises in the middle?
Jane: I didn't see any annoyed faces either ...
...
Jane: ... something weird happened to me today, we went for a typical lunch with my coworkers, but I think I lost my integrity, and complimented a food I did not actually like!
Carl:

***Answers:**
A: I think gift for Poppy made you think of her and made you so happy, that you complimented the chef even though you didn't actually like the food.
B: You? Did you actually complimented a food? you're always so strict and stern about critiquing other people's food. Maybe the day, you know, the yoga, Lucy's traveling and Poppy made you actually enjoy the food. What was it?
C: That's surprising, maybe it's because you were feeling so good ... Anyhow, did Lucy tell you how her backpacking was in Texas?
D: Jane, I think it's understandable to feel like this. But your integrity isn't lost on an occasion like this. ... How do you feel about the situation now?
E: Don't be silly, it does not make you less of a critic. ...

Fig. 1: An example of the framework's templates. [TAG] indicates masks to be replaced with a chosen social group's attributes. Answers and the prompt are only given to the model if cognitive empathy is being evaluated. Otherwise, it generates Carl's last response. Option (A) shows the deepest understanding since it is closest to the causal tuple – three dots in the text mark parts excluded for brevity.

taining the following components in order: (i) $prompt_{CE}$, *context*, *conversation*, *answers* for CE multiple choice, (ii) *context*, *conversation* for AE and ERA. We pass each of the variants to the model separately. The resulting CE score is computed as the accuracy of selecting the choices that manifest the most understanding of the speaker's state across all samples. The AE score comprises Valence, Arousal, and Dominance scores. While we consider them unreliable measures of affective empathy because they measure it indirectly, they can provide valuable information about the type of affect the model uses in its response. For measuring ERA, we use the EPITOME's dimensions IP, EX, and ER.

We evaluate *Llama-3.1-8B* [7] and *Zephyr-gemma-v.1* [33,34]. A 80 GB Nvidia A100 graphics card was used to process the evaluation.

4 Results

Table 1 shows significant differences in all three evaluated dimensions (CE, AE, ERA) measured across the models and templates. The metrics respective to the dimensions are: accuracy of multiple-choice answers (CE), VAD (AE) and EPITOME scores (ERA). For the different intersectional groups, the measured variance between scores is much smaller. Generally, *zephyr-gemma* performed

			Cognitive Empathy	*Affective Empathy*			*Response Appropriateness*		
Model	**Template**	**Count**	**MC↑**	**V↓**	**A↓**	**D↓**	**IP↑**	**EX↑**	**ER↑**
Llama-3.1	0	243	0.29	0.16	0.11	0.13	0.12	1.56	0.54
	1	243	0.35	0.15	0.11	0.13	0.19	0.09	0.91
Zephyr-gemma	0	243	0.01	0.22	0.14	0.18	0.08	0.19	0.79
	1	243	0.10	0.18	0.13	0.16	0.07	0.01	1.60

Table 1: Evaluation of the generated responses on the framework's three dimensions of empathy, the interpretation of the metrics is explained in Section 2. The metric for Cognitive empathy is the accuracy of selecting choices with the most understanding among the multi-choice answers (example in Fig. 1). *Llama-3.1* has higher accuracy in this task. *Llama-3.1* is also more stable across the VAD metrics, and also Empathetic Explorations (EX) on only one of the templates, which were low otherwise. *Zephyr-gemma* had higher scores in Empathetic Responding (ER), this may be caused by the fact that it tends to role-play less.

much worse in CE. The results show that it was easiest for both of the models to find the answer with the most understanding to the sample shown in Figure 1. The differences in AE scores between the models are smaller and *Llama-3.1* achieves lower scores and smaller differences between the two samples, ERA scores are similar.

For pronouns, one of the evaluated dimensions, there are no obvious differences between the scores. We found outliers in the intersection of these groups,

such as Lesbian/She, which has an Interpretations (IP) score significantly above the overall average ($\sigma = 3.36$), but were unable to find any noticeable differences in the generated output.

Both models followed the multi-choice prompt well. For the free generation, manual inspection of a small subset the outputs suggests that *Llama-3.1* might follow role-playing better, *zephyr-gemma* tends respond from the third perspective (10-15 %), or set the situation up in a couple of sentences (10 %), instead of directly responding.

5 Conclusion

We provided a disambiguation of empathy for computational models to help future work define the construct closer to its psychological origins as opposed to the loose definitions that are currently widespread. As a main result, we proposed a new empathy evaluation framework for the responses generated by conversational agents that acknowledges the inherent subjectivity of empathetic understanding. The framework focuses on how empathetic understanding and responding is influenced by intersectional bias. It provides methods to generate evaluation samples from templates by inserting the intersectional contexts into them. The framework uses a new three-dimensional measurement operationalization of empathy to measure the construct. We demonstrated the usage of the framework on a small synthetic sample. In all three framework dimensions, we measured significant differences between the *Llama-3.1-8B* and *Zephyr-gemma-v.1* models. Lastly, we identified differences in empathetic understanding across the evaluated metrics in some intersectional groups. More importantly, we showed the framework's strength in providing the ability to stratify scores across a wide range of social contexts, giving a more fine-grained insight into model behavior and potential harms.

Limitations

We view the modest number of templates as a limitation. Even though we can produce many examples by substituting into the masks, most of their structure stays the same. Further, the contexts and conversations do not reflect the true variety across multiple different groups; future work should thus focus on increasing the number and diversity of the sample creators. The template structure itself, especially the inclusion of context as an explanation of the speaker's background, does place limitations the naturalness and ecological validity of the framework. Lastly, while the number of empathy metrics in this work is limited, future works can use the outlined criteria to include other metrics.

Acknowledgements. This work was supported by the project, Research of Excellence on Digital Technologies and Wellbeing `CZ.02.01.01/00/22_008/0004583` which is co-financed by the European Union.

References

1. Blodgett, S.L., Barocas, S., Daumé III, H., Wallach, H.: Language (technology) is power: A critical survey of" bias" in nlp. arXiv preprint arXiv:2005.14050 (2020)
2. Buechel, S., Buffone, A., Slaff, B., Ungar, L., Sedoc, J.: Modeling empathy and distress in reaction to news stories. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4758–4765. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1507, https://aclanthology.org/D18-1507
3. Chiu, Y.Y., Sharma, A., Lin, I.W., Althoff, T.: A computational framework for behavioral assessment of llm therapists. arXiv preprint arXiv:2401.00820 (2024). https://doi.org/10.48550/arXiv.2401.00820
4. Coll, M.P., Viding, E., Rütgen, M., Silani, G., Lamm, C., Catmur, C., Bird, G.: Are we really measuring empathy? proposal for a new measurement framework. Neuroscience & Biobehavioral Reviews **83**, 132–139 (2017)
5. Debnath, A., Conlan, O.: A critical analysis of empatheticdialogues as a corpus for empathetic engagement. In: Proceedings of the 2nd Empathy-Centric Design Workshop. EMPATHICH '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3588967.3588973, https://doi.org/10.1145/3588967.3588973
6. Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.W., Gupta, R.: Bold: Dataset and metrics for measuring biases in open-ended language generation. In: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency. pp. 862–872 (2021)
7. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
8. Gallegos, I.O., Rossi, R.A., Barrow, J., Tanjim, M.M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., Ahmed, N.K.: Bias and fairness in large language models: A survey. Computational Linguistics pp. 1–79 (2024)
9. Grainger, S.A., McKay, K.T., Riches, J.C., Chander, R.J., Cleary, R., Mather, K.A., Kochan, N.A., Sachdev, P.S., Henry, J.D.: Measuring empathy across the adult lifespan: A comparison of three assessment types. Assessment **30**(6), 1870–1883 (2023)
10. Hassenzahl, M.: Experience design: Technology for all the right reasons. Morgan & Claypool Publishers (2010)
11. Jahoda, G.: Theodor lipps and the shift from "sympathy" to "empathy". Journal of the History of the Behavioral Sciences **41**(2), 151–163 (2005)
12. Lahnala, A., Welch, C., Jurgens, D., Flek, L.: A critical reflection and forward perspective on empathy and natural language processing. arXiv preprint arXiv:2210.16604 (2022)
13. Lee, A., Kummerfeld, J., Ann, L., Mihalcea, R.: A comparative multidimensional analysis of empathetic systems. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 179–189 (2024)
14. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: A manually labelled multi-turn dialogue dataset. arXiv [cs.CL] (Oct 2017)
15. Lipps, T.: Leitfaden der psychologie. W. Engelmann (1909)
16. Maibom, H.L.: Affective empathy. In: The Routledge handbook of philosophy of empathy, pp. 22–32. Routledge (2017)

17. Malhotra, G., Waheed, A., Srivastava, A., Akhtar, M.S., Chakraborty, T.: Speaker and time-aware joint contextual learning for dialogue-act classification in counselling conversations. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 735–745 (2022)
18. Mohammad, S.M.: Word affect intensities. arXiv preprint arXiv:1704.08798 (2017)
19. Morales, S., Clarisó, R., Cabot, J.: A dsl for testing llms for fairness and bias. In: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. pp. 203–213 (2024)
20. Paluck, E.L., Green, S.A., Green, D.P.: The contact hypothesis re-evaluated. Behavioural Public Policy **3**(2), 129–158 (2019)
21. Pérez-Rosas, V., Wu, X., Resnicow, K., Mihalcea, R.: What makes a good counselor? learning to distinguish between high-quality and low-quality counseling conversations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 926–935 (2019)
22. Raj, C., Mukherjee, A., Caliskan, A., Anastasopoulos, A., Zhu, Z.: Breaking bias, building bridges: Evaluation and mitigation of social biases in LLMs via contact hypothesis. arXiv [cs.CL] (Jul 2024). https://doi.org/10.48550/arXiv.2407.02030
23. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Su, J., Duh, K., Carreras, X. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016). https://doi.org/10.18653/v1/D16-1264, https://aclanthology.org/D16-1264
24. Rashkin, H., Smith, E.M., Li, M., Boureau, Y.L.: Towards empathetic open-domain conversation models: A new benchmark and dataset. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5370–5381. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1534, https://aclanthology.org/P19-1534
25. Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., Hashimoto, T.: Whose opinions do language models reflect? In: International Conference on Machine Learning. pp. 29971–30004. PMLR (2023)
26. See, A., Roller, S., Kiela, D., Weston, J.: What makes a good conversation? how controllable attributes affect human judgments. arXiv preprint arXiv:1902.08654 (2019)
27. Sharma, A., Miner, A., Atkins, D., Althoff, T.: A computational approach to understanding empathy expressed in text-based mental health support. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5263–5276. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.425, https://aclanthology.org/2020.emnlp-main.425
28. Sheng, E., Chang, K.W., Natarajan, P., Peng, N.: The woman worked as a babysitter: On biases in language generation. arXiv preprint arXiv:1909.01326 (2019)
29. Sicilia, A., Alikhani, M.: Learning to generate equitable text in dialogue from biased training data. arXiv preprint arXiv:2307.04303 (2023)
30. Sotolar, O., Formanek, V., Debnath, A., Lahnala, A., Welch, C., FLek, L.: Empo: Emotion grounding for empathetic response generation through preference optimization (2024), https://arxiv.org/abs/2406.19071
31. Spaulding, S.: Cognitive empathy. In: The Routledge handbook of philosophy of empathy, pp. 13–21. Routledge (2017)

32. Stade, E., Stirman, S.W., Ungar, L.H., Schwartz, H.A., Yaden, D.B., Sedoc, J., DeRubeis, R., Willer, R., et al.: Artificial intelligence will change the future of psychotherapy: A proposal for responsible, psychologist-led development (2023)
33. Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A.M., Wolf, T.: Zephyr: Direct distillation of lm alignment (2023)
34. Tunstall, L., Schmid, P.: Zephyr 7b gemma. `https://hf.co/HuggingFaceH4/zephyr-7b-gemma-v0.1` (2024)
35. Wan, Y., Wang, W., He, P., Gu, J., Bai, H., Lyu, M.R.: Biasasker: Measuring the bias in conversational ai system. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 515–527 (2023)
36. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? arXiv [cs.AI] (Jan 2018)
37. Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., Du, M.: Explainability for large language models: A survey. arXiv (Sep 2023)
38. Zhu, Q., Chong, L., Yang, M., Luo, J.: Reading users' minds from what they say: An investigation into llm-based empathic mental inference. arXiv preprint arXiv:2403.13301 (2024)

Named Entity Alignment in Czech-English Parallel Data

Zuzana Nevěřilová[1] and Hana Žižková[2]

[1] Natural Language Processing Centre
Faculty of Informatics
Botanická 68a, Brno, Czech Republic 3839@mail.muni.cz
[2] Department of Czech Language
Faculty of Arts
Arna Nováka 1, Brno, Czech Republic zizkova@phil.muni.cz

Abstract. The paper introduces an approach to named entity alignment for Czech-English parallel data. We enriched the Parallel Global Voices corpus with named entity recognition (NER) and named entity linking (NEL) annotations. The new annotation layer employs sentence transformers and cosine similarity to identify NE translations across English, Czech, and possibly other languages, considering atypical entity pairings and achieving an F1 score of 0.94 on evaluated samples.

Keywords: named entity recognition, named entity linking, named entity translation, sentence transformers.

1 Introduction

In our previous work [10], we introduced an efficient method for creating parallel named entity (NE) datasets. We benefited from an existing resource, the Parallel Global Voices [11], and existing models for named entity recognition (NER) annotation. For English, we used the `dslim/bert-large-NER` model from HuggingFace [5,13]. For Czech, we used the Czert-B multi-purpose model [12]. In the project, we aimed to perform named entity linking (NEL) to Wikidata. Finally, we published a dataset where the parallel sentences have another two layers of annotation: 1. NER for classes: PERson, LOCation, ORGanization, and MISCellaneous 2. NEL with links of some English NEs into Wikidata QNames

For the disambiguation of English NEs, we used the OpenTapioca platform [4] with a re-ranking method that uses sentence transformers[3]. In this work, we use the latter for cross-language named entity linking. The goal is to find new links to Wikidata, even for Czech NEs. The ultimate goal is to propose an efficient method for building datasets with NER and NEL annotations from parallel data.

[3] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 59–64, 2024.

2 Related Work

Named entity translation has been e.g., in [1], which proposes an algorithm for translating NEs between English and Arabic. In [2], the authors propose a system for NE translation between English and Arabic. A chunk symmetry strategy and English–Chinese transliteration model are used in [8].

In our case, the translation candidates were found in previous work. The task is more straightforward – to establish NE alignment between appropriate candidates. A similar (but harder) task is aligning English NE annotations with other languages. Transformer models are used, e.g., in [7], for alignment of NEs in German, Spanish, Dutch, and Chinese, with F1 from 0.71 to 0.81.

3 PGV NER Dataset

Parallel Global Voices (PGV [11]) is a massively parallel (756 language pairs), automatically aligned corpus of citizen media stories translated by volunteers. The Global Voices community blog contains several guides, including the Translators' guide[4]. It contains recommendations to "localize" whenever possible. Also, it mentions English as the most significant source language. However, according to authors of the PGV [11], the source language for the translation cannot always be reliably identified. PGV contains texts crawled in 2015, reporting "on trending issues and stories published on social media and independent blogs in 167 countries" [11].

The NER annotation with the existing tools performed well in terms of precision: BERT-large-NER achieved 0.77 precision in the MUC-5 exact evaluation scheme [3], and Czert-B achieved 0.79 precision in the same evaluation scheme. On the other hand, the recall of English and especially Czech models was low: 0.45 and 0.2, respectively, in the MUC-5 exact evaluation scheme. In [10], we set up the annotation task with detailed instructions following the Universal-NER [9] annotation scheme. We proved that manual annotation could be performed relatively efficiently using high-precision/low-recall pre-annotations.

The dataset of NE annotations in the parallel data (the NER ground truth) was published in the LINDAT/CLARIAH-CZ repository[5], together with links to Wikidata (the NEL ground truth) for English NEs. In addition, we published the NER model for Czech based on RobeCzech and trained on CNEC[6].

4 Alignment Implementation

For the parallel sentence pairs with marked NEs, we established the following algorithm that finds the translation:

[4] https://community.globalvoices.org/guide/lingua-guides/lingua-translators-guide/

[5] http://hdl.handle.net/11234/1-5533

[6] https://huggingface.co/popelucha/robeczech-NER

1. encode all NEs into embeddings using sentence transformers[7]
2. calculate cosine similarities matrix for NEs in source and target languages
3. iterate from the most similar pairs:
   (a) if the NEs at positions (i, j) share the same class, establish an alignment $R(i, j)$
   (b) set all similarities in row i and column j with a similarity lower than a threshold to 0
   (c) repeat until the similarity matrix is not a zero matrix

The reason for such an apparently complicated algorithm is that the translations are not necessarily 1:1. As shown in Figure 1, in some cases, a foreign NE is mentioned and translated to Czech in the same sentence. The algorithm does not discard an already used NE; instead, it tries to find other similarities with the NE.

The downside of this approach is that the algorithm cannot distinguish the order of the NEs. Figure 2 shows the annotation corrected manually. The model proposed relations between all mentions of *Nigeria*.

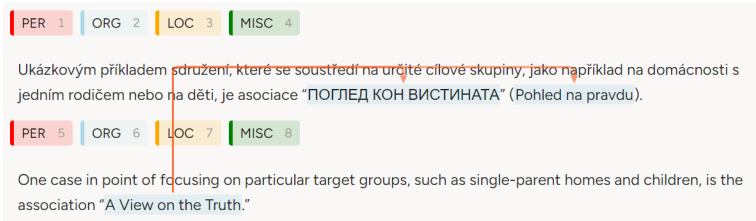


Fig. 1: Example with multiple translations: The organization name is mentioned and translated. The model establishes two alignments.

Fig. 2: Example with multiple NE occurrences: *Nigeria* is mentioned two times. The model can find the translation; however, it finds relations between all occurrences.

[7] `sentence-transformers/distiluse-base-multilingual-cased-v2`

5 Results

We evaluated the proposed alignments against manual annotation. We selected the MUC-5 evaluation scheme[3], although it was not considered to be used for relations. The MUC-5 distinguishes five cases:

- CORrect – the predicted value equals the ground truth
- INCorrect – the predicted value does not equal the ground truth (in NER evaluation, this is used for an incorrect label)
- PARtially correct – in NER, a partial overlap between predicted and ground truth data exist
- MISsed – prediction did not find a value
- SPUrious – prediction found a false positive value

We only considered COR, MIS, and SPU cases where the alignment was established correctly, missed, or added extra, respectively.

We selected a subset of 20 documents for manual annotation. The subset contains 590 sentence pairs, 373 of which contain entities, 320 contain relations. The total numbers of entities are 763 and 684 for sources and target languages, respectively. One NE pair per sentence pair is the most common situation. The results for different similarity thresholds are presented in Table 1. The threshold does not affect the results significantly since, in most cases, there is only one possibility how to align the NEs.

Value	$t = 0.2$	$t = 0.3$	$t = 0.4$	$t = 0.5$	$t = 0.6$	$t = 0.7$	$t = 0.8$	$t = 0.9$
CORrect	569	568	568	568	568	568	567	564
MISsed	43	44	44	44	44	44	45	48
SPUrious	32	33	34	35	37	38	39	40
precision	0.95	0.95	0.94	0.94	0.94	0.94	0.94	0.93
recall	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.92
F1	0.94	0.94	0.94	0.93	0.93	0.93	0.93	0.93

Table 1: No. of missed and spurious alignments for different similarity thresholds t.

5.1 Discussion

Some translation errors originate in the algorithm and its inability to calculate the correct order of NE pairs. This leads to a lower F1 score but does not affect the quality of the translation pairs.

The similarity threshold can be the subject of further experiments. When set too high, similar NEs with different parts of speech can be missed. In addition, the embedding similarity can differ from language to language, i.e., the similarity between an English NE and its Czech translation can be higher than between an English NE and its Macedonian translation.

The similarity threshold was examined, with best values around 0.3. We could see that with lower threshold, the model incorrectly translated similar NEs, such as *Adidas* to *Nike*.

On the other hand, with a reasonable threshold, the model can find NE translations, even if they are incomplete or contain typos. This can be a benefit for the planned task where the NEs in the target language can be found a priori using the embedding similarities between all candidates (see Section 6.1).

5.2 Known Issues

The UniversalNER community does not agree on how to annotate possessives. Since English possessives are (proper) nouns with the *'s*, they are easily considered (proper) nouns within the NER task. On the other hand, Slavonic languages use two competing strategies on how to express possession: genitive noun phrases (e.g., *kniha pana profesora*, *book of the professor* meaning *professor's book*) and possessive adjectives (e.g., *Alicina kniha*, *Alice's book*). As described in [6], adjectives formed from names of human males (with suffix *-ův*) and females (with suffix *-in*) have a paradigm distinct from other adjectives in Czech. The translations found in the dataset are not exact since we translate nouns to adjectives.

A related issue is in translation pairs that contain a noun in English but an adjective in Czech. For example, *pekingská policie* is translated as *Beijing police*. From the NER annotation point-of-view, the Czech expression is not an NE, while *Beijing* is a LOCation. Our method cannot find an NE translation, although the expressions can be translated.

In some cases, we observed the inverse case. For example, *African stories* are translated as *příběhy Afriky* (using a genitive noun phrase). In such cases, the NEs are not translated at all since *African* is an entity of type MISC (similar to national origin). In contrast, *Afrika* (*Africa*) is a LOCation.

6 Conclusion and Future Work

The result of our task is multifold: by another manual annotation, we can identify and correct annotation errors. We plan to release a new version of the NER-NEL dataset. We also plan to incorporate Czech NEL via the English NEL and alignments.

6.1 Finding NEs in Unannotated Data

The procedure (embedding similarity) can be used for unknown data. We plan to experiment with English-Czech sentences without the Czech NER annotation. The question is how well the model can find Czech NEs. Instead of known NEs, we can use all n-grams from the sentence as NE candidates and let the model select the best ones.

If successful, this method could be used for language pairs with no NER model for the target language.

Acknowledgements. This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Al-Onaizan, Y., Knight, K.: Translating named entities using monolingual and bilingual resources. In: ACL. pp. 400–408 (2002), http://www.aclweb.org/anthology/P02-1051.pdf
2. Awadallah, A., Fahmy, H., Hassan Awadalla, H.: Improving named entity translation by exploiting comparable and parallel corpora (2007), https://www.microsoft.com/en-us/research/publication/improving-named-entity-translation-exploiting-comparable-parallel-corpora/
3. Chinchor, N., Sundheim, B.: MUC-5 Evaluation Metrics. In: Fifth Message Understanding Conference (August 1993), https://aclanthology.org/M93-1007
4. Delpeuch, A.: OpenTapioca: Lightweight Entity Linking for Wikidata (2020)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
6. Janda, L., Townsend, C.: Czech. Languages of the world: Materials, Lincom Europa (2000), https://books.google.cz/books?id=6VkbAQAAIAAJ
7. Li, B., He, Y., Xu, W.: Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment (2021), https://arxiv.org/abs/2101.11112
8. Li, P., Wang, M., Wang, J.: Named entity translation method based on machine translation lexicon. Neural Computing and Applications **33**(9), 3977–3985 (May 2021)
9. Mayhew, S., Blevins, T., Liu, S., Šuppa, M., Gonen, H., Imperial, J.M., Karlsson, B.F., Lin, P., Ljubešić, N., Miranda, L., Plank, B., Riabi, A., Pinter, Y.: Universal NER: A Gold-Standard Multilingual Named Entity Recognition Benchmark (2024)
10. Nevěřilová, Z., Žižková, H.: Named Entity Linking in English-Czech Parallel Corpus. In: E. Nöth, A. Horák, P.S. (ed.) TSD 2024. pp. 147–158. Springer International Publishing, Switzerland (2024). https://doi.org/http://dx.doi.org/10.1007/978-3-031-70563-2_12
11. Prokopidis, P., Papavassiliou, V., Piperidis, S.: Parallel Global Voices: a Collection of Multilingual Corpora with Citizen Media Stories. In: Proc. of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 900–905. European Language Resources Association (ELRA), Portorož, SI (May 2016), https://aclanthology.org/L16-1144
12. Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., Konopík, M.: Czert – Czech BERT-like Model for Language Representation. In: Mitkov, R., Angelova, G. (eds.) Proc. of the International Conference on Recent Advances in NLP (RANLP 2021). pp. 1326–1338. INCOMA Ltd., Held Online (Sep 2021), https://aclanthology.org/2021.ranlp-1.149
13. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419

Annotating Health Records: Does Ground Truth Even Exist?

Kristof Anetta

Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic
xanetta@fi.muni.cz

Abstract. This paper introduces a new ground truth subset of the CSEHR dataset, a dataset of Czech health records annotated using a schema of 14 classes that is an adapted version of Apache cTAKES Core Clinical Element types. The paper details the considerations involved in (re)defining individual annotation classes in attempts to maximize utility in computational understanding of medical text.

Keywords: Czech, Electronic health records, EHR, annotation, named entity recognition, NER, medical concept mining.

1 Introduction

The development of medical entity recognition models requires an annotated health record dataset, and the annotation of a health record dataset requires a well-chosen annotation schema. This paper will present some of the considerations encountered during work on augmenting the CSEHR dataset [3], a recent annotated dataset of Czech oncology health records from the Masaryk Memorial Cancer Institute in Brno, Czech Republic.

The initial version of the CSEHR dataset involved 11 student annotators with varying degrees of precision and recall (with respect to hypothetical perfection in following the annotation manual). Given the constraints of time

Table 1: Statistics of CSEHR dataset with added ground truth.

	Original student data	New ground truth
Records	168	20
Sentences	3,566	801
Words	49,530	12,547
Tokens	69,699	19,095
Percentage of tokens annotated	40.6%	53.5%
Number of tokens annotated	28,329	10,218
Total number of annotation tags	31,610	11,556

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 65–74, 2024.

and resources, the annotation quality was sufficient for the dataset to become a promising base for bootstrapping larger datasets and training larger models, and thus be treated as "truth". Nevertheless, the inherent imperfections of student annotation prevented the evaluation from being anything more than n-fold cross-validation. For a better bootstrapping protocol, we were missing a "truer truth" that would diminish the propagation of annotation faults.

Hence, our next step was to develop an addition to the CSEHR dataset, an improved section, the creation of which involved full, one-person focus on consistency and density, homogenizing annotation rule interpretations and avoiding omissions. Just like in the case of student annotation, the BRAT annotation tool [7] was used, together with automated scripts for fixing errors. The comparison of the student portion of the dataset and the newly developed ground truth can be found in Table 1.

2 Annotation schema

The new ground truth was annotated using the same 14-class schema as the larger student annotation section, one based on the 6 core clinical elements [1]

Table 2: 14 entity categories used for annotating the CSEHR dataset, compared with Apache cTAKES types themselves and other notable annotation schemas: Zhu et al. [10], CLEF [5], and i2b2 [8]. Dark rectangles mark the overlap of classes above and below.

Apache cTAKES	Our annotation schema	Zhu et al.	CLEF	i2b2
DiseaseDisorder	**DiseaseDisorder**	Disease	Condition	Medical Problem
SignSymptom	**SignSymptom**	Symptom		
Medication	**Medication_name**	Drug	Drug	Treatment
	Medication_strength			
	Medication_dosage			
Procedure	**Procedure**	Treatment	Intervention	
			Investigation	
Lab	**Lab_name**	Test		Test
	Lab_value		Result	
	Lab_unit			
AnatomicalSite	**AnatomicalSite_name**	Body	Locus	
			Sub-location (modifier)	
	AnatomicalSite_laterality		Laterality (modifier)	
	Negation		Negation (modifier)	
	DateTime			
	Abbreviation			
		Personal History		
		Equipment		
		Department		

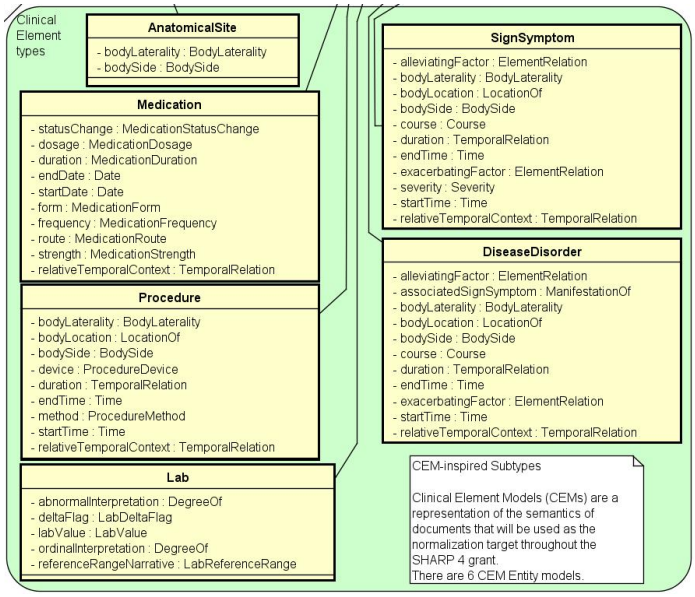


Fig. 1: Core entities in the type system of Apache cTAKES [1].

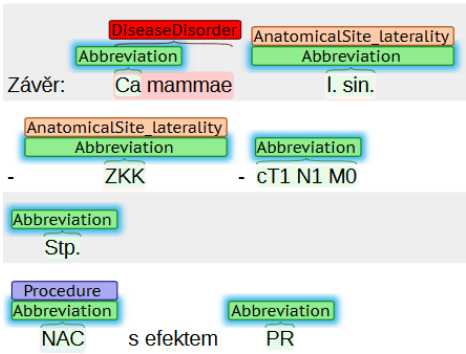
(Figure 1) in the type system of Apache cTAKES [6], an open-source NLP system for the extraction of clinical information from free text. The complete list of tags in this adapted schema can be seen in Table 2, next to comparable annotation schemas in related literature. The non-medical categories *Abbreviation*, *DateTime*, and *Negation* were added for practical reasons.

3 Related work in designing annotation schemas

Although the Apache cTAKES [6] type system is just one of many attempts to efficiently represent salient information in medical text, its intuitions are very similar to the efforts of other teams and organizations. Zhu et al. [10], after reviewing a number of existing medical corpora, synthesized an annotation schema of *Disease*, *Symptom*, *Test*, *Treatment*, *Drug*, and *Body*, which corresponds almost exactly to that of cTAKES, with the addition of classes for *Personal History*, *Equipment*, and *Department*.

The CLEF corpus [5] from 2007 also employed an annotation schema with significant overlaps, as shown in Table 2.

Perhaps the most well-known schema, that of the 2010 i2b2 challenge [8], has only three types (*Medical Problem*, *Treatment*, and *Test*) in its annotation guideline [2]. Although the granularity is markedly different, these three general types subsume most of the other schemas' more specific types.




```
<p>
<s>
Závěr     klgInSc1 závěr-n  závěr
<g/>
:         kIx,     :-x      :
Ca        kA       Ca-x     ca       Carcinoma                                          B-DiseaseDisorder             B-Abbreviation
mammae    klgFnPc2 mammae-n mammae                                                     I-DiseaseDisorder
l         kA       l-x      l        vlevo                                             B-AnatomicalSite_laterality   B-Abbreviation
<g/>
.         kIx.     .-x      .        vlevo                                             I-AnatomicalSite_laterality   I-Abbreviation
sin       kA       sin-x    sin      vlevo                                             I-AnatomicalSite_laterality   I-Abbreviation
<g/>
.         kIx.     .-x      .        vlevo                                             I-AnatomicalSite_laterality   I-Abbreviation
-         kIx-     --x      -
ZKK       kA       ZKK-x    ZKK      zevní kvadranty                                   B-Abbreviation                B-AnatomicalSite_laterality
-         kIx-     --x      -
cT1       k4       cT1-m    cT1      klinická klasifikace tumor 1, nodus 1, metastáze 0  B-Abbreviation
N1        klgMnSc1 N1-n     N1       klinická klasifikace tumor 1, nodus 1, metastáze 0  I-Abbreviation
M0        klgMnSc1 M0-n     M0       klinická klasifikace tumor 1, nodus 1, metastáze 0  I-Abbreviation
</s>
</p>
<p>
<s>
Stp       k?       Stp-x    Stp      stav po                                           B-Abbreviation
<g/>
.         kIx.     .-x      .        stav po                                           I-Abbreviation
</s>
<s desamb="1">
NAC       kA       NAC-x    NAC      neoadjuvantní chemoterapii                        B-Abbreviation                B-Procedure
s         k7c7     s-p      s
efektem   klgInSc7 efekt-n  efekt
PR        k0       pr-i     pr       parciální remise                                  B-Abbreviation
</s>
</p>
```

Fig. 2: Example of annotation including nested annotations shown both in BRAT and in BIO format.

4 Considerations

4.1 Coverage of the schema

There are several different possible goals one can have in mind when annotating health records. They include, but are not limited to:

1. **Precise retrieval of decision-crucial classes**, such as looking for disease names.
   - Advantages: Sensitivity to a limited number of high-impact classes, easier to create training data.
   - Disadvantages: Most of the detailed information is ignored.
2. **Complete ontological knowledge of the text**, which aims to be able to categorize almost every word.
   - Advantages: Most of the text is represented on multiple levels, searchable, suitable for relation extraction.
   - Disadvantages: Very high number of classes, ambiguous classes, difficult annotation and training, lower precision.

3. **Extraction of structured information**, which puts emphasis on the discrete and continuous variables hidden in plain text.
   - Advantages: Often, structured information has well-defined orthographical forms and available values.
   - Disadvantages: There are too many possible structured information slots pertaining to a patient. Unless we know what information we are looking for, we are unlikely to get it.

Currently, the annotation schema of CSEHR cuts somewhere between 1 and 2. Its focus on 6 core clinical elements gets salient information about both the patient (*DiseaseDisorder*, *SignSymptom*, *AnatomicalSite*) and how they are treated (*Procedure*, *Medication*, *Lab*). However, the schema is missing a significant portion of health record text simply because much of the information does not fall under the 6 clinical elements. The following section details these gaps in representation.

4.2 Unrepresented concepts

Notably, the gaps occur in:

- Specifying adjectives.
  `jódového` (*iodine* (*adjective*))
  `vstřebatelné` (*absorbable*)
  `alveolární` (*alveolar*)
  `závažných` (*severe*)
  `objemné` (*voluminous*)
  are only a small sample of unannotated medically relevant properties of concepts.
- Nouns of medical objects and devices.
  `(jódové) zrno` ((*iodine*) *grain*)
  `stehy` (*stitches*)
  `RTG 3 GE Discovery XR 656` (*machine name*)
  all remain outside the annotation schema because, while related to a *Procedure*, they are only its means or results. In post hoc analysis, we found that an addition of a *Procedure_device* class would cover this gap and it will be considered for the next iterations of the dataset.
- Narrow domain concepts, in this case highly specialized oncological and genetic sections
  `cT2N1histolog.M0` (*cancer staging notation*)
  This spaceless blend of codes and an abbreviated word refers to clinical TNM (tumor-nodus-metastasis) staging of cancer: tumor size stage 2, nodular involvement stage 1 with histological confirmation, no metastases. The closest annotations suitable for the string might be those of the *Lab* class expanded for more complex observations, if the string were broken down into single-letter units of meaning.
  `NBN/c.657_ 661del/p.LysAsnfs*16` (*genetic mutation*)

This is a detailed specification of a genetic mutation. Its closest annotation in the schema might be *SignSymptom* as it is a description of a mutation which is an observed sign, although that does not do justice to the specific information contained therein.

4.3 Definitional conundra per clinical element

AnatomicalSite Theoretically, an anatomical site can be as big as the

`trup` (*trunk*)

and as small as a

`gen` (*gene*)

Is a

`uzlina` (*nodule*)

an *AnatomicalSite* when it can refer to any nodule in the human body? Are adjectives like

`plicní` (*pulmonary*)

sufficient references to an anatomical site to be annotated as such? In the ground truth annotation, we put the practical boundary of *AnatomicalSite* so that only body parts visible to the human eye (or visible if surgically removed) are annotated as such (hence including the nodule) and adjectives are not considered *AnatomicalSite*s.

Lab Interpreted strictly, the pair of *Lab_name* and *Lab_value* (*Lab_unit* optional) only refer to laboratory measurements related to the patient, such as

`P_Natrium: 143 mmol/l` (*id.*)

However, the regular orthographical structure of this class naturally offers an expansion to more general body measurements not determined in a laboratory, e.g.

`BMI: 25.28` (*id.*)
`TK/puls: 123/ 90/ 88` (*blood pressure*/*heart rate*)

As long as the values remain numeric, this expansion seems clearly suitable for training NER models.

The definition of *Lab_value*s as numeric might be too constraining if we want to capture a variety of observation values. The *Lab* class is especially promising for the extraction of structured information by slot filling, the *Lab_name* corresponding to a slot to be filled with *Lab_value*. Therefore, this paper will delve into a more extensive analysis.

Some laboratory results have a string for value

`HER2 negativní` (*HER2 negative*)

and, together with the non-laboratory measurements mentioned above, they open an avenue for an entire continuum of examination observations to fall into the *Lab* class. This ambiguous space stretches from fairly measurement-like entries such as

`porody: 2` (*childbirths: 2*)

to statements of more or less specified presence or absence of a condition

`thyreopatie 0` (*thyreopathy 0 = no thyreopathy*)

to discrete observations of reported behavior

`kouření: nyní` (*smoking: currently*)

At this point, if all of the above are allowed to be annotated as *Lab*, the class begins to stretch dangerously, resulting in three main issues:

1. Almost every copular clause linking a complement to a medically relevant subject becomes a candidate for the *Lab* class:
   `mamy symetrické` (*mammae symmetrical*)
   `břicho nebol.` (*abdomen non-tender*)
2. The dilemma of multiple values. Some measurements and observations yield multiple values.
   `Břicho v niveau, měkké, prohmatné, palpačně nebolestivé, bez rezistence, neperitoneální, poklep dif. bubínkový, peristaltika v normě` (*Abdomen at level, soft, palpable, non-tender on palpation, without resistance, non-peritoneal, percussion diffusely tympanic, peristalsis normal*)
   When should such a structure stop being treated as a name-value(s) pair? In our annotation schema, as long as the list of observations after a *Lab_name* is contiguous, they may all be annotated as *Lab_value*s.
3. In instances of hierarchically structured observations such as
   `Hlava : poklepově nebolestivá, výstupy trigeminu nebolest., zornice izokorické` (*Head: non-tender to percussion, trigeminal nerve exits non-tender, pupils isocoric*)
   in one perspective,
   `Hlava` (*Head*)
   is the *Lab_name* meaning "examination of the head" and all items that follow are *Lab_value*s, as they are all observations related to the head. At the same time,
   `zornice` (*pupils*)
   is an even more precise *Lab_name*, meaning "status of pupils during head examination"
   `izokorické` (*isocoric*)
   being its *Lab_value*. Hence,
   `zornice` (*pupils*)
   would be both a *Lab_name* and a part of a greater *Lab_value*, and
   `izokorické` (*isocoric*)
   would be *Lab_value* twice, once completely and once partially.

DiseaseDisorder In cases like

> `středně diferencovaný duktalni invazivni karcinom prsni zlazy s metastatickym postižením axillarni lymfaticke uzliny` (*moderately differentiated ductal invasive carcinoma of the breast with metastatic involvement of the axillary lymph node*)

it is difficult to determine the span of the *DiseaseDisorder* annotation. In both directions starting from

> `karcinom` (*carcinoma*)

the words are adding specification, but the more specification is added, the harder it becomes to use the whole cluster for NER model training.

For the needs of ontological representation, a reasonable solution is to find the most specific corresponding entry in the ICD [9]. For language model training, the preferred way would be that of splitting up the concept into core disease identifier and attached properties.

Medication While searching for literal occurrences of medication names, the *Medication_name* class is one of the easiest to annotate - reliable databases of medications exist even in countries without other medical vocabularies.

However, the question is if the class should include concepts

- broader than a specific medicine: names of medication classes such as
  `diuretika a Ca blokátoru` (*diuretic and calcium channel blocker*)
  or even
  `medikace` (*medication*)
  itself,
- or narrower: names of active substances or chemical constituents in the medication.

5 Discussion

Annotated health records are a valuable resource in healthcare computing, but they are most valuable when created with a clear purpose reflected in the annotation schema. To answer the question from the title: there is never a single Medical Annotation Truth, but rather many local, purpose-bound utilitarian truths.

Though a good benchmark, the Apache cTAKES Core Clinical Element schema is designed with a realistic industry application in mind: to cherry-pick prominent occurrences of unambiguous concepts with high accuracy and link them directly to a UMLS (Unified Medical Language System) [4] dictionary entry. In the case of the *Lab* class, to extract measurement values in sufficiently regular form.

Researchers interested in finding an annotation schema to make large datasets searchable and filterable by the occurrence of key medical concepts

need look no further - the *DiseaseDisorder*, *SignSymptom*, *Procedure* and *Medication* alone can be expected to satisfy an overwhelming majority of search queries.

For researchers interested in a complete medical-ontological representation of texts and at the same time a class count reasonably low for model training, a new annotation schema should be developed, aiming specifically at high coverage and low class count. If the cTAKES types were to be adapted for this purpose, they would require some broadening (e.g. *Lab* to *Observation*) and a small number of new classes that cover medical contexts and body properties that remained outside the schema in this paper, perhaps even a class to mark narrow-domain information.

6 Conclusion

In this paper, we introduced the new ground truth subset of CSEHR and presented examples of decision making during annotation, especially considerations related to class definitions, class boundaries, and gaps in representation. The unexpected frequency of encountered ambiguities and gaps is a strong reminder of the need of careful planning and utilitarian class design in medical text annotation for language model training.

Acknowledgements. The analyzed Czech data was kindly provided by the Masaryk Memorial Cancer Institute in Brno, Czech Republic.

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. Apache cTAKES - User FAQs — svn.apache.org. `https://svn.apache.org/repos/infra/websites/production/ctakes/content/user-faqs.html`, [Accessed 09-11-2023]
2. Concept annotation guidelines, `https://www.i2b2.org/NLP/Relations/assets/Concept%20Annotation%20Guideline.pdf`
3. Anetta, K., Horák, A.: New human-annotated dataset of czech health records for training medical concept recognition models. In: International Conference on Text, Speech, and Dialogue. pp. 110–120. Springer (2024)
4. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research **32**(suppl_1), D267–D270 (01 2004). https://doi.org/10.1093/nar/gkh061, `https://doi.org/10.1093/nar/gkh061`
5. Roberts, A., Gaizauskas, R., Hepple, M., Davis, N., Demetriou, G., Guo, Y., Kola, J.S., Roberts, I., Setzer, A., Tapuria, A., et al.: The clef corpus: semantic annotation of clinical text. In: AMIA Annual Symposium Proceedings. vol. 2007, p. 625. American Medical Informatics Association (2007)
6. 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)

7. 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)
8. Uzuner, Ö., South, B.R., Shen, S., DuVall, S.L.: 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. Journal of the American Medical Informatics Association **18**(5), 552–556 (2011)
9. World Health Organization: ICD-10 : International statistical classification of diseases and related health problems : Tenth revision (2004)
10. Zhu, E., Sheng, Q., Yang, H., Liu, Y., Cai, T., Li, J.: A unified framework of medical information annotation and extraction for chinese clinical text. Artificial Intelligence in Medicine **142**, 102573 (2023)

Part III

Text Corpora

Impact of Data Split and Vocabulary Size in Neural Machine Translation for Slovak Language

Matúš Kleštinec

Constantine the Philosopher University Nitra
Trieda Andreja Hlinku 1, 949 01 Nitra, Slovakia
matus.klestinec@ukf.sk

Abstract. In this paper, we focus on machine translation, specifically its process from obtaining texts to evaluating the machine translation model. Our machine translation models were trained from source language English to target language Slovak. We explored the influence of various factors on the machine translation model, such as the size of the test and validation sets, vocabulary size, the impact of tokenization algorithms used, text quality and size of bilingual texts. After training the machine translation model, we also examined the effects of translation parameters on the result text. Evaluation was done using automatic metrics such as BLEU, METEOR, and COMET, as well as manual inspection of sentences. We found that the parameters we investigated had an impact on the machine translation model and optimal settings for our models.

Keywords: Machine Translation, Corpus, Slovak.

1 Introduction

Machine translation is the automated process of translating sentences from one natural language to another using computers [1]. A machine translation model trained with data, in this case, sentences in the desired languages, is capable of evaluating the most suitable words in the target language, thus creating a translation in the desired language. There are not many extensive and high-quality parallel corpora available that include Slovak language, so the optimal preprocessing and training settings have not been thoroughly explored yet. In this work, we present few possible settings for hyperparameters in the preprocessing and training steps for translating, from English to Slovak language.

2 Related Work

Other author [2] have also used the OpenNMT toolkit for training neural machine translation. The author [2] specifically used the OpenNMT-lua variant and trained on the EUR-Lex dataset in the English-Czech language pair, also tokenized text using Byte Pair Encoding. His neural machine translation models

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 77–83, 2024.

achieved BLEU scores 61.07 and 62.02. Authors [3] experimented with vocabulary size of models, which were trained on low-resource languages. For training, they used Akkadian, Manipuri and Lower Sorbian languages. They used automatic metrics such as BLEU and COMET for evaluation and their results showed that, for some languages, its better to have smaller vocabulary size. Another benefit is faster training time, because of smaller vocabulary. In conclusion they concluded, that optimal vocabulary size depends on language.

3 Methodology

For experimentation purposes and to account for available computational resources, we selected a relatively small parallel corpus, Europarl version 7 [4], which contains 640715 sentences in English and Slovak. Europarl corpus is already aligned, so this step was not required. During preprocessing, we applied the following steps:

- Unicode normalization,
- Removal of identical sentences: source sentence = target sentence,
- Removal of duplicate lines,
- Removal of lines that are too long,
- Filtering of text using langdetect library,
- Additional removal of certain characters resulting from previous steps,
- Transformation of text to lowercase,
- Reordering of the lines.

One step requires more detailed explanation: text filtering using the langdetect library. Filtering rows with the Python langdetect library is a useful tool when texts contain words or entire sentences that do not match the desired language or languages. Langdetect processes n-grams of the chosen text and outputs the probability that the text belongs to a particular language [5]. This way, we can filter out sentences that do not belong to English or Slovak.

By applying these preprocessing methods, we adjusted the corpus to the desired form and removed 29236 sentence pairs. Number of sentence pairs left was 611479. We tokenized the corpus into subwords using the Byte Pair Encoding algorithm for most of the models, except one, on which we used Unigram algorithm and split the corpus into training, validation, and testing sets. We trained the models using the OpenNMT-py toolkit [6], which utilizes the Transformer architecture [7]. We evaluated the models using the automatic metrics BLEU [8], METEOR [9], and COMET [10] (specifically COMET-22 [11]). The code for our solution can be found on GitHub [12].

4 The Impact of the Data Split Ratio for Training, Testing, and Validation Sets on Model Quality

We trained models V2 through V5 and model V10. The evaluation results of each model can be seen in Table 1, where the individual models are ordered by

the size of the testing and validation sets. The Test/valid value represents the number of sentences allocated separately for each set. Values highlighted in red represent the worst result within the respective column, while green represents the best result. Our scales for values BLEU, METEOR and COMET is between 0 and 1.

Table 1: Comparison of models with different testing and validation set sizes.

Model	BLEU	METEOR	COMET	Test/Valid	Test/Valid
V10	0.39361	0.6744	0.8978	2000	≈ 0.33
V5	0.39908	0.7038	0.8993	4000	≈ 0.66
V2	0.39889	0.6765	0.9008	6000	≈ 0.98
V3	0.39959	0.6890	0.9009	30000	≈ 4.91
V4	0.40082	0.4942	0.8992	60000	≈ 9.81

As you can see in Table 1, we divided the testing and validation set from approximately 0.33% to 9.81%, with the upper limit representing the standard split of 80/10/10% for training, testing, and validation sets. When comparing models based on BLEU scores, a significant difference appears only between model V10 and the other models, with V10 having a lower score by 0.00528 compared to V2 and by 0.00193 compared to V4. In the case of METEOR scores, the models maintain consistent values, except for model V4, which has a significantly different score from the others. The difference between model V4 and V10 is 0.1802, representing 18%, and is statistically significant. The COMET metric shows smaller differences, with model V10 again having the lowest rating, but it is closer to the other models. The difference between V10 and V3 is only 0.0031, representing 0.31%. The results from all metrics indicate that a small testing and validation set is inadequate, as model V10 achieved the lowest ratings in BLEU and COMET metrics, while METEOR had the second lowest rating.

Based on these findings in Table 1, we believe that the optimal split for the testing and validation sets should be approximately 0.66% to 4.91%, which we can round to 5%. This range is indicative and may not apply to all machine translation models.

5 The Impact of Vocabulary Size on Model Quality

We experimented with vocabulary sizes ranging from 2000 to 50000. Number of training steps was set to 100000. Table 2 details the models with their respective metrics, vocabulary size, and the number of training steps. The models are sorted by vocabulary size. Since we used the same vocabulary size for both the source and target languages, we only use one column labeled ”Vocabulary Size” to represent this value.

Table 2: Comparison of models with different vocabulary sizes.

Model	BLEU	METEOR	COMET	Vocabulary Size	Number of Training Steps
V12	0.39867	0.7843	0.9024	2000	100000
V9	0.42028	0.6905	0.9058	4000	100000
V11	0.45228	0.6905	0.9131	8000	100000
V6	0.46227	0.6932	0.9147	16000	80000
V7	0.47777	0.7937	0.9147	32000	55000
V8	0.44686	0.7351	0.9194	50000	35000

As the vocabulary size increased, the number of training steps decreased as you can see in Table 2. Models with a vocabulary size exceeding 8000 led to early termination of training through the early stopping condition. Models with a very small vocabulary, such as V12 and V9, had the worst BLEU scores, while models with increasing vocabulary sizes (V11, V6, and V7) achieved better results, with the highest score observed for model V7 with a vocabulary of 32000. On the other hand, the largest model, V8, showed a decrease of 0.03091 in BLEU score, likely due to overfitting. For the COMET metric, scores increased with vocabulary size, but without significant jumps the difference between the worst model, V12, and the best model, V8, was only 0.0170. METEOR scores, however, were inconsistent; model V12, which had the lowest BLEU and COMET scores, achieved the second highest METEOR score, while models V9 and V11 had the lowest METEOR scores. The largest vocabulary size did not yield the best METEOR score either, with a noticeable difference between models V7 and V8 (0.0586).

If we look at how the translated sentences from each model look, we will find that the differences are not that significant. We can demonstrate on one sentence, how different was translation for each model.

Sentence from source language: In the current situation the european union should pay particular attention to its 20 million small and mediumsized enterprises

Reference sentence from target language: V súčasnej situácii by mala európska únia osobitnú pozornosť venovať svojim 20 miliónom malých a stredných podnikov

Translation of model V12: V súčasnej situácii by mala európska únia venovať osobitnú pozornosť svojim 20 miliónom malých a stredných podnikov

Translation of model V9: V súčasnej situácii by európska únia mala venovať osobitnú pozornosť 20 miliónom malých a stredných podnikov

Translation of model V11: V súčasnej situácii by európska únia mala venovať osobitnú pozornosť 20 miliónom malých a stredných podnikov

Translation of model V6: V súčasnej situácii by európska únia mala venovať osobitnú pozornosť 20 miliónom malých a stredných podnikov

Translation of model V7: V súčasnej situácii by európska únia mala venovať osobitnú pozornosť 20 miliónom malých a stredných podnikov

Translation of model V8: V súčasnej situácii by európska únia mala venovať osobitnú pozornosť 20 miliónom malých a stredných podnikov

All sentences maintained their context after translation and are comparable to the reference sentence, with some differences in the order of certain words. Except for model V12, all models produced the same translation for the sentence, but this may not represent the entire set of translated sentences. We selected only one sentence from 4,000 sentences. Manually comparing all sentences would be time consuming.

Based on all metrics, model V12 can be considered the worst and model V7 the best in Table 2. However, the optimal vocabulary size is not universal. It depends on the corpus size and content. The experiment showed that a small vocabulary can negatively impact translation quality, while an excessively large vocabulary may not be beneficial. Further experimentation is necessary to find the optimal value for our specific model.

6 Discussion

Creating a neural machine translation model was also conducted by the author [13]. Besides standard preprocessing steps, such as removing of punctuations, we used langdetect library. This allowed us to filter out 9390 lines of text that were not in either Slovak or English. For training purposes, the author [13] also used OpenNMT and worked with vocabulary sizes of 10000 and 100000. In our approach, we experimented with various vocabulary sizes ranging from 2000 to 50000, which led to significantly higher BLEU scores. For comparison, the BLEU score for the models trained by the author [13] with a vocabulary size of 10000 was 0.0604, and 0.1217 for a vocabulary size of 100000. We converted the BLEU scores presented by the author [13] to our 0-1 scale for consistency. Our model with a vocabulary size of 2000, which was the lowest performer in our comparison in Table 2, achieved a BLEU score of 0.39867. The number of training steps also differed, as most of our models were trained for 100000 steps and validated every 2500 steps, unlike the author's models, which were trained for 20000 steps and validated every 1000 steps. Most notable difference is that author used 2 layer LSTM architecture, while we used 6 layer Transformer architecture.

In this work, we introduced machine translation and its significance, the steps necessary to achieve a functional machine translation model, and addressed the settings for training and translation parameters.

Based on experiments with the size of the test and validation set distribution, we found that the most suitable division ranges from approximately 0.66% to 4.91%. These values are indicative, representing an approximate distribution that yielded the best results. It is important to note that for a larger model, this range could differ. We worked with a relatively small corpus, making it impossible to experiment with a larger number of sentences with the same quality.

Experiments with vocabulary size demonstrated that vocabulary size can significantly influence the model. An excessively small vocabulary can negatively impact the final model, but the same holds true for an excessively large vocabulary. Based on our findings, we concluded that it is always necessary to consider the ideal vocabulary size, which may vary for each solution. We only experimented with English and Slovak, using one corpus size, so our values apply only to this pair of languages and for an approximate corpus size. The languages have different lexical richness, which could also affect the appropriate vocabulary size for the machine translation model. A larger corpus could contain a richer vocabulary, which would also influence the suitable vocabulary size.

By comparing models with different tokenization algorithms, we reached the conclusion that Byte Pair Encoding yields better results than the Unigram algorithm. The experiment with tokenization algorithms was conducted on a small sample, as we compared only two models. While the sample is small, all metrics, especially METEOR, indicate a decline in translation quality.

The numerous possibilities make it both time consuming and computationally demanding to test everything, which can be considered the greatest limitation of this work. Testing all possible combinations of settings would be impractical, so our results cover only a small portion of potential solutions for improving machine translation. Another limitation is the relatively small amount of quality bilingual texts in English and Slovak. While Europarl is a high quality corpus, its quite small.

Acknowledgements. This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-23-0554.

Disclosure of Interests. The author have no competing interests to declare that are relevant to the content of this article.

References

1. Tan, Z. Wang, S. Yang, Z. Chen, G. Huang, X. Sun, M. Liu, Y.: Neural machine translation: A review of methods, resources, and tools. In: AI Open, Volume 1, pp. 5-21. (2020) https://doi.org/10.1016/j.aiopen.2020.11.001
2. Wörgötter, B.: Domain-specific English-Czech Neural Machine Translation, `https://is.muni.cz/th/k8nt8/` (2018)
3. Signoroni, E. Rychlý, P.: Better Low-Resource Machine Translation with Smaller Vocabularies. In: Nöth, E., Horák, A., Sojka, P. (eds) Text, Speech, and Dialogue. TSD 2024. Lecture Notes in Computer Science(), vol 15048. Springer, Cham. (2024) `https://doi.org/10.1007/978-3-031-70563-2_15`
4. 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. (2012)
5. Langdetect, `https://github.com/fedelopez77/langdetect?tab=readme-ov-file`, last accessed 2024/02/29

6. Klein, G. KIM, Y. DENG, Y. SENELLART, J. RUSH, A.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. In: Proceedings of ACL 2017, System Demonstrations, pp. 67-72. Association for Computational Linguistics, Vancouver, Canada (2017)
7. Vaswani, A. Shazeer, N. Parmar, N. Uszkoreit, J. Jones, L. Gomez, A. Kaiser, L. Polosukhin, I.: Attention Is All You Need. In: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000-6010. Curran Associates Inc, 57 Morehouse LaneRed HookNYUnited States (2017)
8. Papineni, K. Roukos, S. Ward, T. Zhu, W.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 311-318. Association for Computational Linguistics, Philadelphia, USA (2002)
9. BANERJEE, S. LAVIE, A.: METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp 65-72. Association for Computational Linguistics, Ann Arbor, Michigan (2005)
10. REI, R. STEWARD, C. FARINHA, A. 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 (2020)
11. REI, R. SOUZA, J. ALVES, D. ZERVA, C. FARINHA, A. GLUSHKOVA, T. LAVIE, A. COHEUR, L. MARTINS, A.: COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In: Proceedings of the Seventh Conference on Machine Translation (WMT), pp. 578-585. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022)
12. GitHub repository, `https://github.com/ukf-matusklestinec/Strojovy-preklad-DP`
13. Pavlišin, P.: Slovenský neurónový strojový preklad pomocou knižnice OpenNMT, `https://opac.crzp.sk/?fn=detailBiblioForm&sid=2F8FD603D177BA679037F0795407` (2022)

A Comparative Study of Text Retrieval Models on DaReCzech

Jakub Stetina[1], Martin Fajcik[1], Michal Štefánik[2], and Michal Hradis[1]

[1] Faculty of Information Technology
Brno University of Technology, Czech Republic
[2] Faculty of Informatics
Masaryk University, Czech Republic
{xsteti05,ifajcik,ihradis}@fit.vutbr.cz, stefanik.m@mail.muni.cz

Abstract. This article presents a comprehensive evaluation of 7 off-the-shelf document retrieval models: Splade, Plaid, Plaid-X, SimCSE, Contriever, OpenAI ADA and Gemma2 chosen to determine their performance on the Czech retrieval dataset DaReCzech. The primary objective of our experiments is to estimate the quality of modern retrieval approaches in the Czech language. Our analyses include retrieval quality, speed, and memory footprint. Secondly, we analyze whether it is better to use the model directly in Czech text, or to use machine translation into English, followed by retrieval in English. Our experiments identify the most effective option for Czech information retrieval. The findings revealed notable performance differences among the models, with Gemma22 achieving the highest precision and recall, while Contriever performing poorly. Conclusively, SPLADE and PLAID models offered a balance of efficiency and performance.

Keywords: Information Retrieval, Evaluation, Comparison, Czech Language, Performance Assessment, Document Retrieval, Model Analysis.

1 Introduction

Information retrieval (IR) is used in areas such as search engines and question-answering systems. Lately, we've seen advancements in IR models [12,32,16, *inter alia*], but picking the right one for a non-English document collection can be challenging. We address this gap for Czech language by doing a comprehensive comparison in our study. In particular, we utilize DareCzech, a Czech retrieval and ranking dataset [14], for testing IR models to evaluate different IR models on Czech documents and queries. Our contributions are: (1) we analyze the index sizes to understand the storage requirements of various models, (2) we analyze the retrieval speed of such methods, to estimate how these models scale to large corpora in their default implementation, (3) we conduct ranking performance testing using multiple metrics on off-the-shelf models, and (4) we compare different model types, including those tested directly on the Czech

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 85–100, 2024.

dataset as well as on an English translation of the Czech dataset, keeping in mind the respective model's training data language, to provide insights into different approaches for indexing and retrieving czech. To the best of our knowledge, this is the first comparison study of existing state-of-the-art retrieval methods in the Czech language.

2 Related Work

Several well-known benchmarks have been used for evaluating information retrieval (IR) and text embedding models. **MS MARCO** [2] is widely used for passage and document retrieval, offering real-world web queries and answers. **MIRACL** [33] is a multilingual benchmark designed for retrieval across different languages. **MTEB** [18] provides a comprehensive evaluation across diverse tasks, including clustering, classification, and re-ranking.

Beyond English-language benchmarks, several datasets focus on IR evaluation within specific linguistic contexts. For Czech, the **CWRCzech** dataset [28] includes 100M query-document pairs based on Czech click data from *Seznam.cz* search logs. German-language IR is explored through **DPR German** [19] and **German LEGAL IR** [29], which assess retrieval in general and legal domains. The **SKQuad** [17] dataset provides an IR benchmark specifically for the Slovak language and the **Scandinavian Embedding Benchmark** (**SEB**) [7] provides a comprehensive evaluation framework for text embeddings in Scandinavian languages. Within MTEB, Polish [23] and Chinese [30] datasets extend the evaluation to language-specific IR tasks.

The dataset utilized in our study is **DaReCzech** (see Subsection 4.1), introduced in [14], which is specifically tailored for the Czech language and consists of manually annotated query-document pairs. The relevance annotations in DaReCzech are not binary, allowing for a more nuanced evaluation of relevance ranking models and enabling the use of various evaluation metrics.

3 Model Descriptions

BM25[24]. BM25 is a traditional lexical approach which has been widely used and had been the standard before the rise of neural models. It ranks documents based on a query's term frequency, inverse document frequency, and document length, meaning the importance of each term in the query and document is considered along with the document's length normalization, to produce relevance scores for each document.

In our study, we employed a BM25 baseline to assess the effectiveness of the other models. This model stands out as the only model with lemmatization applied to the query and document content, a distinction arising from the nature of BM25, which is not a neural model and relies on the precise lexical form of terms within the corpus.

For the Czech language, which features word inflection, lemmatization is essential for precise term matching and relevance ranking. Therefore, the lemmatized version of the corpus for BM25 is required.

splade-cocondenser-ensembledistil (SPLADE) [9]. (Sparse Lexical and Expansion Model) leverages sparse vocabulary-sized representations to leverage the advantages of BOW (bag-of-words). Splade operates by first applying a linear transformation to the BERT [6] output embeddings, then performing a dot product with the token embeddings from the whole vocabulary, resulting in a matrix of scores where each input token has a score for every token in the vocabulary[3]. While its predecessor, SparTerm [1], used a learned binary mask to select relevant scores from this matrix, Splade induces sparsity through a combination with a FLOPS regularizer and a logarithmic function during the representation computation. The final representation is obtained by summing the weights along the sequence tokens, producing a sparse embedding with a dimensionality equal to the vocabulary size. The second version of Splade improves this pooling mechanism to instead use the max for each token from the vocabulary. The model used in this comparison is the highest performing distilled version of splade as described in [8].

ColbertV2.0 (PLAID) [25]. The PLAID model represents a multi-vector approach. It extends the late interaction mechanism used in **ColBERT** [13] to enhance efficiency in information retrieval. In the original version of ColBERT, the comparison of query-document embeddings was performed by matching every token in the document embedding with every token in the query embedding, calculating scores using a maximum similarity function, where the highest similarity value for each query token across all document tokens is retained. ColBERTv2 [26] improved upon this approach by clustering the document embeddings into centroid clusters, thereby enabling a more efficient retrieval process. At search time, a fixed number of candidate clusters are selected, and their embeddings are decompressed to compute the final similarity scores.

In addition, **PLAID**, further enhances efficiency and performance by introducing a multi-stage candidate generation process. This approach includes steps for pruning and centroid-based interactions, progressively narrowing down the set of candidate passages. The final, smaller set of potential passages is then scored, resulting in a more streamlined and scalable retrieval pipeline. The model used in this study was trained on the English MS MARCO.

Plaidx-xlmr-large-mlir-neuclir (PLAID-X) [20,31]. Multilingual version of PLAID, PLAID-X builds upon the ColBERT architecture and employs a multilingual encoder XLM-RoBERTa (XLM-R) for multilingual and cross-lingual encodings. The model used in this study was trained using the translate-train approach on Chinese, Persian, and Russian data, relying on the XLM-R encoder for cross-language mappings.

[3] It utilizes the already pretrained masked language modeling head.

Text-embedding-ada-002 (OpenAI ADA) [22]. A closed-source model, that uses cosine similarity to compare two embeddings to calculate the resulting score.

Contriever-msmarco (Contriever) [10]. Contriever is a dense retrieval model that leverages self-supervised contrastive learning to effectively learn representations for information retrieval tasks. The model distinguishes between positive and negative passage pairs, where positive pairs are generated through independent window cropping from the original context (a document), and random token deletion. These approaches ensure that positive pairs share semantic content while exhibiting variation in phrasing, but also occasionally retaining lexical overlap. Negative pairs, on the other hand, are mined using MoCo[11], a method that builds a queue and uses a slowly changing encoder to generate negative samples. The model updates its document encoder by incorporating an online average of past parameters, ensuring that representations remain consistent across nearby training steps, hence making old representations stored in the queue compatible. This self-supervised training approach enables Contriever to produce dense vector representations for queries and documents, facilitating efficient retrieval and robust generalization across various retrieval tasks without the need for labeled datasets. The model used in this study was further finetuned on the English MS MARCO [2].

Simcse-dist-mpnet-paracrawl-cs-en (SimCSE)[10,4]. SimCSE employs contrastive learning to generate sentence embeddings, using simple dropout-based noise to create positive pairs from the same sentence while drawing negative pairs from other sentences within the batch. This approach trains the model to capture semantic similarities and differences between sentences without relying on large supervised datasets. Positive pairs are formed through augmentation techniques, such as random token deletion, replacement and masking, which introduce variability while preserving the original meaning. We chose to specifically test a model trained on the ParaCrawl [3] dataset (`SimCSE-Dist-MPNet-ParaCrawl`) as it achieved the highest DaReCzech performance at P@10 in the original work. The model used in this study was pretrained using an undisclosed Czech dataset from Seznam.cz and distilled on `czeng20-csmono` [15] and `Paracrawler v9` [3].

BGE Multilingual Gemma2 (Gemma2) [5]. BGE-Multilingual-Gemma2 is a large-language model based multilingual embedding model. It is directly finetuned using contrastive objective on an undisclosed diverse set of languages and tasks based on `google/gemma-2-9b` model [27]. During evaluation, the prompt used was: "Given a web search query, retrieve relevant passages that answer the query," as outlined in the instructions [4].

[4] `https://huggingface.co/BAAI/bge-multilingual-gemma2`

4 Experimental Setup

4.1 DaReCzech Dataset

DaReCzech is a Czech dataset designed for text relevance ranking, comprising over 1.6 million query-document pairs. It is divided into Train-big (1.4M pairs for model training), Train-small (97K pairs for model training), Dev (41K pairs), and Test (64K pairs), with no overlap between splits. Each record includes a query, URL, document title, document body text extract (BTE), and a relevance label. Queries are real user inputs, with minor corrections, and the documents are preprocessed to exclude irrelevant sections, ensuring a cleaner representation of content for ranking tasks.

We utilized DaReCzech by selecting test queries along with their associated relevant documents and additional documents to create a 100,000-document sample for indexing. This approach allowed us to maintain a representative document pool without indexing the entire dataset, primarily due to computational and economical overhead. Specifically, for the OpenAI Ada model, embedding generation incurs a cost. This balanced approach enabled a comprehensive evaluation while managing resource expenditure effectively. For relevance scores, we classified documents with scores above 0 as relevant and those with a score of 0 as non-relevant, aligning with binary metrics like precision and recall. More about the evaluation criteria can be found in Appendix A.1.

4.2 BM25 grid search

For fair comparison, we ran a grid search on the development set within our corpus to find the most optimal setting of the BM25's hyperparameters. The performance of the BM25 model was most optimal when the document length normalization parameter B was set to its maximum value of 1.0. This adjustment highlighted the importance of document length normalization in our particular case. The K_1 parameter, saturation of term frequency, showed minimal impact on our corpus, suggesting that its tuning had little to no effect on the performance. Based on this experiment, the BM25 hyperparameters were set to $[K_1, B] = [2, 1]$.

4.3 Dataset Translation

Some of the models tested were primarily or exclusively trained on English data. To achieve optimal performance and ensure a fair comparison, we applied document-level translation to the DaReCzech corpus, translating it into English using OPUS-MT, a multilingual translation model based on the OPUS corpora [5]. This approach allowed us to evaluate all models in their supported language setting.

[5] OPUS-MT translation model: `https://huggingface.co/Helsinki-NLP/opus-mt-cs-en`.

4.4 Segmentation

For the purpose of our evaluation, we employed a common indexing methodology across all models. For an initial experiment, we indexed documents in two ways: using only a truncated section up to each model's maximum input limit, and as multiple overlapping segments for longer documents, thus running the evaluation with two separate indices for each model[6]. However, since the overlapping approach did not yield any significant improvement, as can be seen in Figures 1 and 2, the later tested models were evaluated using non-overlapping segments only. The overlapping segments revealed an inherent bias of DaReCzech, as all the important data were usually concentrated at the beginning of the documents. This was also indicated by our extra analysis in Appendix C, making the cutoff method with no overlap sufficient. The cutoff lengths for each model were derived from the respective model papers, and a stride (if used) was selected to be roughly one-third of the maximum token length for each model as can be seen in Table 1.

Table 1: Overview of Experimental Model Configurations. All models were tested using truncated inputs, with select models additionally tested using segmented document inputs. The Segmenting column specifies the maximum token length and overlap window used for these experiments. OpenAI Ada is an only closed-access model we tested. (*The SPLADE output dimension represents the average number of tokens present in the output vector.).

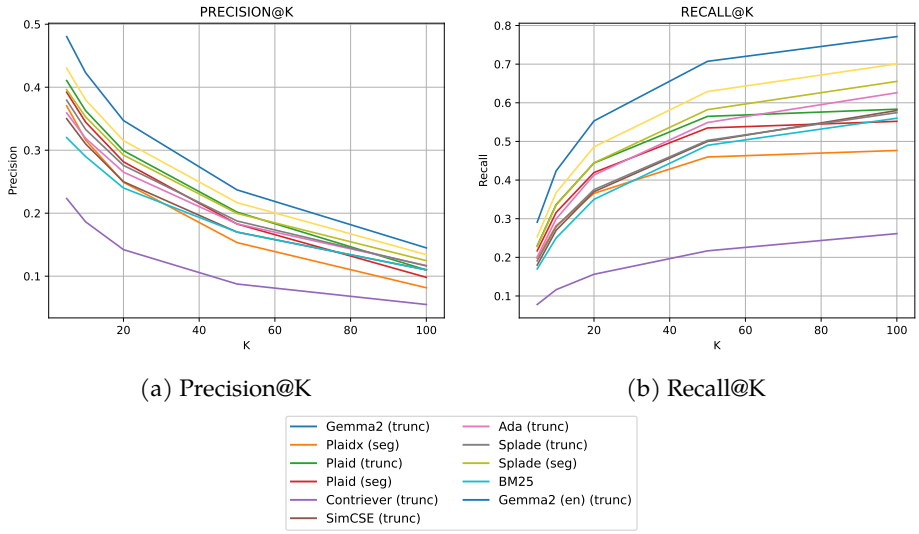
Model	Output Dim.	Max Tokens	Lang.	Segmenting
Splade	*45.7	256	en	256/86
PLAID	128	300	en	300/100
PLAID-X	128	180	cs	180/60
Contriever	768	256	en	–
SimCSE	256	128	cs	–
Gemma2	3584	512	cs/en	–
OpenAI ada	1536	8192	cs	–

5 Results and Analysis

The precision and recall values (Figures 1a,1b) for all models across different k values exhibit distinct patterns, with Contriever performing the worst, even below BM25. The stricly top-performing model is Gemma2. Notably, both the Czech and English versions of Gemma2 rank highly, with the Czech model

[6] As a result, the evaluation with overlapping made the effective number of document represenations in the (single-vector) indices exceed the base count of 100,000 documents. During the retrieval process, all the duplicate versions of documents were removed leaving only the highest rank of the same document.

showing a slight advantage in performance. Beneath Gemma2, the best results come from the PLAID models. However, segmenting the documents with these models demonstrates a decline in both precision and recall as k increases, possibly due to an accumulation of irrelevant information from segment-level retrievals impacting the overall ranking quality[7].

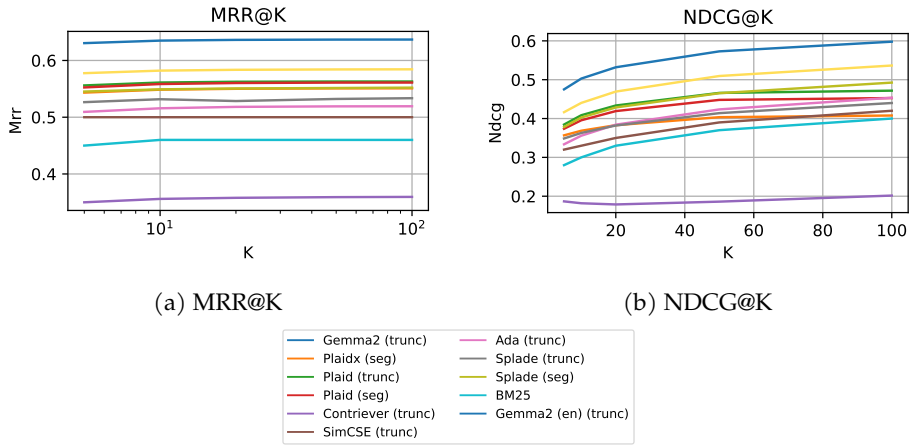


(a) Precision@K (b) Recall@K

Fig. 1: Comparison of Precision and Recall at different values of k.

These observed trends are even more evident in the full MRR and NDCG metrics (Figures 2a 2b), where the differences among models are more pronounced. In the MRR graphs, nearly consistent performance across different k values indicates that while the general retrieval ability remains stable, the ranking quality of results varies significantly between models. The superior performance of Gemma2 and the relative weaknesses of Contriever are reflected here, reinforcing the patterns observed in the precision and recall figures. This alignment suggests that models with higher precision and recall also exhibit better ordering and ranking capabilities, as demonstrated by their MRR and NDCG scores.

[7] The evaluation was conducted in two ways for some models. In one the retrieval function retrieve(k) was called for each tested value of k separately and in the other the highest tested value of k was chosen and then the results cut off down to the tested k value. This was done to especially examine the PLAID retrieval implementation, which determines different hyperparameters for its approximate nearest-neighbor search for different k value. This however did not show any significant change in the tested metrics (in these cases the better results were kept).



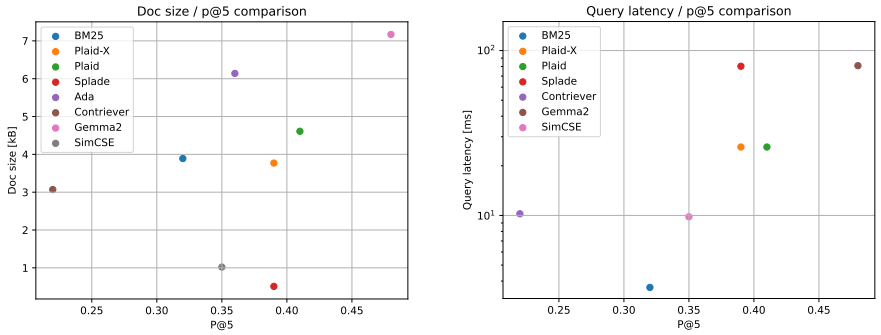
(a) MRR@K (b) NDCG@K

Fig. 2: Comparison of MRR, NDCG at different values of k.

Figure 3a demonstrates the trade-off between document size (estimated by averaging the size of each index by the number of indexed documents) and retrieval precision. As anticipated, BM25 maintains a compact document representation but exhibits a low Precision@5 performance, with Contriever faring even worse. PLAID-X achieves modest gains over BM25, with a smaller index size per document due to a restrictive 180-token limit. SPLADE, while comparable to PLAID-X in precision, maintains a much smaller index thanks to its sparse nature [8]. The original PLAID model, without cross-lingual settings, slightly outperforms both PLAID-X and SPLADE, though it incurs a larger index size due to a higher token limit of 300. OpenAI's Ada model struggles to compete, hindered by its large embedding dimension, resulting in a substantial index size that does not justify its middling performance. The Gemma2 model emerges as the top performer, albeit with the largest embedding size, indicating a trade-off between high retrieval accuracy and storage requirements. Such result is aligned with observations in [21], where authors demonstrate that embedding performance tends to scale with model size and embedding dimension.

An analysis of query latency in Figure 3b shows that BM25 achieves the fastest query times, which aligns with its straightforward term-matching approach. Contriever and SimCSE, both using single-vector embeddings and cosine similarity, follow closely. The PLAID-X and PLAID models exhibit slightly longer latencies, likely due to their multi-stage retrieval process, which involves candidate selection and more complex ranking steps, contributing to a moderate increase in query time. SPLADE and Gemma2 are slower still; SPLADE's

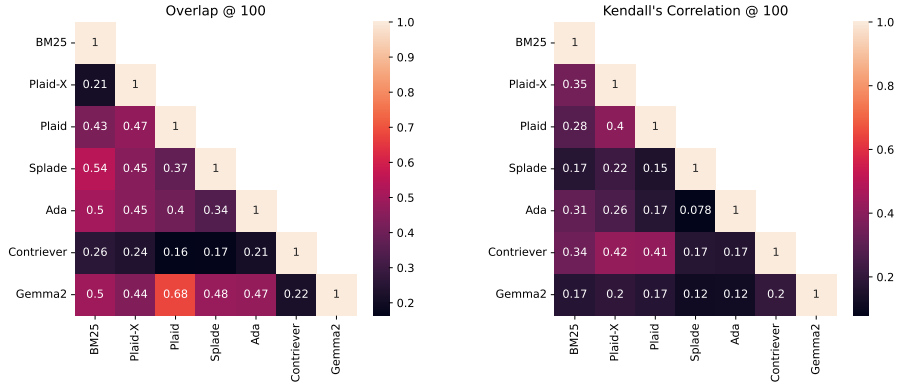
[8] Some models achieve similar or even significantly smaller index sizes compared to BM25 due to truncation; BM25 indexed entire documents without truncation, while many other models were limited to a few hundred tokens per document. This limit, especially for longer documents, led to reduced overall index sizes.



(a) Doc repr. size / P@5 comparison (b) Query latency / P@5 comparison

Fig. 3: Doc size, query latency in relation to P@5.

sparse representation requires additional computation to dynamically calculate sparse scores, while Gemma2's high-dimensional embeddings impose added processing overhead. These patterns suggest that models with multi-stage or complex scoring mechanisms naturally incur higher latency compared to more direct embedding or term-based approaches.



(a) Overlap @ 100 (b) Kendall's Correlation @ 100

Fig. 4: Pairwise overlap and correlation of overlapped items in top-100 responses of different IR systems.

Regarding the overlap among the models, as depicted in Figure 4, Contriever consistently exhibits the lowest overlap scores across comparisons with other models, a finding that aligns well with its previously observed underperfor-

mance in Figures 1a and 1b. Notably, PLAID and PLAID-X display a high degree of overlap and strong Kendall τ correlation, likely attributable to their shared architecture and training approach, with PLAID-X being a multilingual adaptation of PLAID. Interestingly, we also observe a notably high overlap value between PLAID and GEMMA, which could be attributed to GEMMA's training on diverse multilingual data that likely includes features common to PLAID's retrieval methodology.

6 Conclusion

In this paper, we evaluated 7 off-the-shelf information retrieval models on the DaReCzech corpus, comparing their performance against the traditional BM25 approach. The goal was to identify the most effective model for information retrieval in Czech.

Our findings showed that Gemma2 consistently delivered the best precision and recall metrics across various k values, with the Czech version slightly outperforming the English one. However, its high retrieval accuracy came with a large index size due to high-dimensional embeddings exceeding even the multi-vector models. In contrast, BM25 and Contriever exhibited the poorest performance, with Contriever notably underperforming and struggling to match BM25's baseline.

SPLADE and the PLAID models offered a balance between performance and efficiency. SPLADE's sparse representation resulted in the smallest index size, making it suitable for resource-constrained applications, while the PLAID models, especially the original, provided higher precision with modest increases in index size. The ColBERT-based models performed well unsegmented, but segmenting for long documents led to a decrease in performance as k increased.

For Czech-language IR tasks, Gemma2 is recommended if accuracy is the top priority and storage is manageable. SPLADE is a practical choice when memory efficiency is crucial, and PLAID/PLAID-X offer a middle ground, particularly with token limit adjustments. This study underscores the trade-offs between model complexity, storage, and retrieval quality, guiding suitable model selection for Czech-language IR.

Acknowledgements. This work was supported by project Ministry of Culture of the Czech Republic through NAKI III project semANT, grant. no DH23P03OVV060, Horizon EU programme through project ELOQUENCE, grant no. 101135916, and by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bai, Y., Li, X., Wang, G., Zhang, C., Shang, L., Xu, J., Wang, Z., Wang, F., Liu, Q.: Sparterm: Learning term-based sparse representation for fast text retrieval (2020), https://arxiv.org/abs/2010.00768
2. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: Ms marco: A human generated machine reading comprehension dataset (2018), https://arxiv.org/abs/1611.09268
3. Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M.L., Kamran, A., Kirefu, F., Koehn, P., Ortiz Rojas, S., Pla Sempere, L., Ramírez-Sánchez, G., Sarrías, E., Strelec, M., Thompson, B., Waites, W., Wiggins, D., Zaragoza, J.: ParaCrawl: Web-scale acquisition of parallel corpora. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4555–4567. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.417, https://aclanthology.org/2020.acl-main.417
4. Bednář, J., Náplava, J., Barančíková, P., Lisický, O.: Some like it small: Czech semantic embedding models for industry applications (2023), https://arxiv.org/abs/2311.13921
5. Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation (2024), https://arxiv.org/abs/2402.03216
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019), https://arxiv.org/abs/1810.04805
7. Enevoldsen, K., Kardos, M., Muennighoff, N., Nielbo, K.L.: The scandinavian embedding benchmarks: Comprehensive assessment of multilingual and monolingual text embedding (2024), https://arxiv.org/abs/2406.02396
8. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: From distillation to hard negative sampling: Making sparse neural ir models more effective (2022), https://arxiv.org/abs/2205.04733
9. Formal, T., Piwowarski, B., Clinchant, S.: Splade: Sparse lexical and expansion model for first stage ranking (2021), https://arxiv.org/abs/2107.05720
10. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings (2022), https://arxiv.org/abs/2104.08821
11. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning (2020), https://arxiv.org/abs/1911.05722
12. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding (2021), https://arxiv.org/abs/2009.03300
13. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over BERT. CoRR **abs/2004.12832** (2020), https://arxiv.org/abs/2004.12832
14. Kocián, M., Náplava, J., Štancl, D., Kadlec, V.: Siamese bert-based model for web search relevance ranking evaluated on a new czech dataset (2021), https://arxiv.org/abs/2112.01810
15. Kocmi, T., Popel, M., Bojar, O.: Announcing czeng 2.0 parallel corpus with over 2 gigawords. arXiv preprint arXiv:2007.03006 (2020)

16. Koto, F., Li, H., Shatnawi, S., Doughman, J., Sadallah, A.B., Alraeesi, A., Almubarak, K., Alyafeai, Z., Sengupta, N., Shehata, S., Habash, N., Nakov, P., Baldwin, T.: Arabicmmlu: Assessing massive multitask language understanding in arabic (2024), https://arxiv.org/abs/2402.12840
17. Technical University of Košice, D.o.E., Communication, M.: Retrieval-skquad dataset (2023), https://huggingface.co/datasets/TUKE-KEMT/retrieval-skquad, dataset for Slovak search retrieval evaluation, licensed under CC BY-NC-SA 4.0
18. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: Mteb: Massive text embedding benchmark (2023), https://arxiv.org/abs/2210.07316
19. Möller, T., Risch, J., Pietsch, M.: Germanquad and germandpr: Improving non-english question answering and passage retrieval (2021), https://arxiv.org/abs/2104.12741
20. Nair, S., Yang, E., Lawrie, D., Duh, K., McNamee, P., Murray, K., Mayfield, J., Oard, D.W.: Transfer learning approaches for building cross-language dense retrieval models. In: Proceedings of the 44th European Conference on Information Retrieval (ECIR) (2022), https://arxiv.org/abs/2201.08471
21. Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J.M., Tworek, J., Yuan, Q., Tezak, N., Kim, J.W., Hallacy, C., et al.: Text and code embeddings by contrastive pre-training. arXiv preprint arXiv:2201.10005 (2022)
22. OpenAI: Openai ada model for retrieval. https://platform.openai.com/docs/models/embeddings (2023), accessed: 2024-10-29
23. Poświata, R., Dadas, S., Perełkiewicz, M.: Pl-mteb: Polish massive text embedding benchmark (2024), https://arxiv.org/abs/2405.10138
24. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval **3**, 333–389 (01 2009). https://doi.org/10.1561/1500000019
25. Santhanam, K., Khattab, O., Potts, C., Zaharia, M.: Plaid: An efficient engine for late interaction retrieval (2022), https://arxiv.org/abs/2205.09707
26. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: Colbertv2: Effective and efficient retrieval via lightweight late interaction (2022), https://arxiv.org/abs/2112.01488
27. Team, G., Riviere, M., Pathak, S., Sessa, P.G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C.L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J.B., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C.A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozińska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucińska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J.P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., yeong Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonell, K., Nguyen, K., Sodhia, K., Greene, K., Sjoesund, L.L., Usui, L., Sifre, L., Heuermann, L., Lago, L., McNealus, L., Soares, L.B., Kilpatrick, L., Dixon, L., Martins, L., Reid, M., Singh, M., Iverson, M., Görner, M., Velloso, M., Wirth, M., Davidow, M., Miller, M., Rahtz, M., Watson, M., Risdal, M., Kazemi, M., Moynihan, M., Zhang, M., Kahng, M., Park, M., Rahman, M., Khatwani, M., Dao, N., Bardoliwalla, N., Devanathan, N., Dumai, N., Chauhan, N., Wahltinez, O., Botarda,

P., Barnes, P., Barham, P., Michel, P., Jin, P., Georgiev, P., Culliton, P., Kuppala, P., Comanescu, R., Merhej, R., Jana, R., Rokni, R.A., Agarwal, R., Mullins, R., Saadat, S., Carthy, S.M., Cogan, S., Perrin, S., Arnold, S.M.R., Krause, S., Dai, S., Garg, S., Sheth, S., Ronstrom, S., Chan, S., Jordan, T., Yu, T., Eccles, T., Hennigan, T., Kocisky, T., Doshi, T., Jain, V., Yadav, V., Meshram, V., Dharmadhikari, V., Barkley, W., Wei, W., Ye, W., Han, W., Kwon, W., Xu, X., Shen, Z., Gong, Z., Wei, Z., Cotruta, V., Kirk, P., Rao, A., Giang, M., Peran, L., Warkentin, T., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Sculley, D., Banks, J., Dragan, A., Petrov, S., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Borgeaud, S., Fiedel, N., Joulin, A., Kenealy, K., Dadashi, R., Andreev, A.: Gemma 2: Improving open language models at a practical size (2024), https://arxiv.org/abs/2408.00118

28. Vonásek, J., Straka, M., Krč, R., Lasonová, L., Egorova, E., Straková, J., Náplava, J.: Cwrczech: 100m query-document czech click dataset and its application to web relevance ranking. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2024, vol. 38, p. 1221–1231. ACM (Jul 2024). https://doi.org/10.1145/3626772.3657851, http://dx.doi.org/10.1145/3626772.3657851
29. Wrzalik, M., Krechel, D.: GerDaLIR: A German dataset for legal information retrieval. In: Aletras, N., Androutsopoulos, I., Barrett, L., Goanta, C., Preotiuc-Pietro, D. (eds.) Proceedings of the Natural Legal Language Processing Workshop 2021. pp. 123–128. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.nllp-1.13, https://aclanthology.org/2021.nllp-1.13
30. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., Nie, J.Y.: C-pack: Packed resources for general chinese embeddings (2024), https://arxiv.org/abs/2309.07597
31. Yang, E., Lawrie, D., Mayfield, J., Oard, D.W., Miller, S.: Translate-distill: Learning cross-language dense retrieval by translation and distillation. In: Proceedings of the 46th European Conference on Information Retrieval (ECIR) (2024), https://arxiv.org/abs/2401.04810
32. Yüksel, A., Köksal, A., Şenel, L.K., Korhonen, A., Schütze, H.: Turkishmmlu: Measuring massive multitask language understanding in turkish (2024), https://arxiv.org/abs/2407.12402
33. Zhang, X., Thakur, N., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., Liu, Q., Rezagholizadeh, M., Lin, J.: Making a miracl: Multilingual information retrieval across a continuum of languages (2022), https://arxiv.org/abs/2210.09984

A Evaluation Process

A.1 Evaluation Metrics

To assess the performance of these IR models, we employ a range of standard evaluation metrics:

Precision Precision quantifies the accuracy of relevant documents in the retrieved set and is defined as:

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \tag{1}$$

Recall Recall measures the ability of the model to retrieve all relevant documents from the corpus and is given by:

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents in corpus}\}|} \tag{2}$$

MRR (Mean Reciprocal Rank) Mean Reciprocal Rank (MRR) evaluates the ranking quality by taking the mean of the reciprocal ranks of the first relevant document for each query:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{3}$$

where rank_i is the position of the first relevant document for the i-th query and $|Q|$ is the total number of queries.

MAP (Mean Average Precision) Mean Average Precision (MAP) calculates the average precision for each query and averages these scores across all queries, thereby reflecting the model's ranking consistency. For a given query q, the average precision is:

$$\text{AP}_q = \frac{1}{|\{\text{relevant documents for } q\}|} \sum_{k=1}^{N} \text{Precision}(k) \cdot \text{rel}(k) \tag{4}$$

where N is the total number of documents, $\text{Precision}(k)$ is the precision at rank k, and $\text{rel}(k)$ is a binary indicator of relevance at rank k. MAP is then:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \text{AP}_q. \tag{5}$$

nDCG (Normalized Discounted Cumulative Gain) Normalized Discounted Cumulative Gain (nDCG) evaluates the ranked list's quality by considering the position of relevant documents in the ranking. For a query q, nDCG at rank p is calculated as:

$$\text{DCG}_p = \sum_{k=1}^{p} \frac{2^{\text{rel}(k)} - 1}{\log_2(k+1)} \tag{6}$$

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \tag{7}$$

where $\text{rel}(k)$ is the relevance score of the document at rank k, and IDCG_p is the ideal DCG, obtained by sorting documents in the perfect order of relevance.

A.2 Additional Evaluation Criteria

In addition to the standard metrics, we also consider two specific criteria:

Representation Size Examining the memory footprint required by each model's document representations (measured per document as kB/doc),

Query Latency Query latency refers to the duration taken by an information retrieval system to retrieve and present relevant documents in response to a given query.

Kendall's τ Rank Correlation and Lexical Overlap assessing the consistency of ranking across the models on the top 100 retrieved results for each query - helps understanding how well the models agree on the most relevant documents

B BM25 Hyperparameter Tuning

We perform a BM25 grid search to tune the K1 and B parameters for optimal results on the corpus. The results from the grid search are visualized in Figure 5.

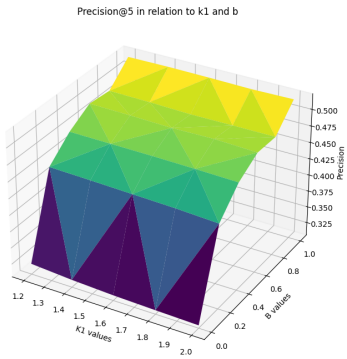


Fig. 5: BM25 hyperparameters grid search.

C ColBERT's Token-level Focus

To estimate which parts of the document are important, the study analyzed ColBERTv2, a model that uses a multi-vector approach, where each token in a document is represented by a separate vector. By examining the vectors of the retrieved documents, tokens with the most interaction with the query were identified. This might indicate which specific parts of the documents were most

relevant to the query and contributed to the retrieval process for the given document.

This experiment examined two modes for the document-query scores (samples with scores aggregated from Colbert's similarity matrix through max-pooling over query token representations (in contrast with the original Colbert's MaxSim operation which computes max-pooling over document token representations) can be seen below in Figure 6) visualized as a probability distribution using the softmax function with the brighter color denoting higher similarity score):

MaxSim operation: particularly chosen as it reflects how the model selects positive document-query pairs, and identifies the best score for each query token with the highest scoring document token, highlighting the most significant interactions.

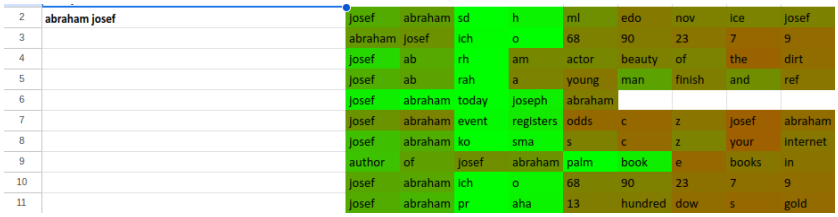


Fig. 6: Colbert interpretability.

A New Czech Pipeline in Sketch Engine

Vlasta Ohlídalová and Miloš Jakubíček

Lexical Computing, Brno, Czechia
Faculty of Informatics, Masaryk University, Brno, Czechia

Abstract. This paper introduces a new Czech pipeline that is now available in Sketch Engine. It describes the tools used for this pipeline and for some of them, we add details of how they were altered in recent years. The most complex part discusses adjustment of the training data used for Czech language – the DESAM corpus – and its effect on accuracy of the POS tagging performed by `RFTagger`.

Keywords: Morphological analysis, corpora annotation.

1 Introduction

For people working with language corpora (linguists, lexicographers, terminologists, translators, ...), POS tagging and lemmatization is the most basic feature that they are utilizing daily. Sketch Engine, as one of the leading corpus managers, does indeed offer those for Czech corpora. However, the quality is not always as good as one would expect. For that reason, years after the previous version, a new version of Czech pipeline was introduced this year. The updated pipeline is based on a more recent version of the Majka [10] morphological database and uses the `RFTagger` [7] instead of the `desamb` tagger [9][1].

The core part of this contribution consists of semi-manual changes to the training data (the DESAM corpus [6]) so that the corpus matches the current version of the Majka morphological database.

2 Majka

Majka is a Czech morphological database containing 3,393,080 wordform + lemma + tag triplets, which are made of 903,888 distinct word forms and 46,000 lemmas. It comes together with a fast (about 1 million words per second) a fast morphological analyzer that queries the database.

A project of building a new Czech dictionary [3] has been running for the last year. The first stage – choosing the lemmas that should be included in the dictionary – is almost finished, so we have used this opportunity to compare the items in the dictionary with the Majka lexicon.

There are altogether 61,676 lemmas approved for the dictionary at the moment (including 407 MWEs). Out of these, approximately 4.5K were not

[1] The POS tagging evaluation comparing desamb and `RFTagger` can be found in [1].

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 101–107, 2024.

found in Majka lexicon, which makes around 7.4 % of all headwords. A part of it (13.5 %) were abbreviations, either written all with capital letters (10 %) or with a dot (3.5 %).

Without these, 3,950 headwords were left as candidates for new words that could be added to the lexicon. Table 1 shows some of the most common words according to frequency list from csTenTen23 [2].

Table 1: Selected words that were accepted in the Czech dictionary project, but they are missing in the Majka lexicon (ordered by frequency).

online	834843
info	164701
Wi-Fi	114165
blog	108700
on-line	104178
vs	98442
iPhone	82655
nej	69862
CO2	68886
fitness	67087
wellness	62956
off	60052
play-off	47187
D1	44251
e-shop	42226
elektro	35803
profi	32209
naživo	32056
wifi	31959
dovolatel	31883
live	31123
make-up	30680
insolvenční	30104

Some of these words can be added to Majka semi-automatically by finding the most similar already existing entries based on their suffix and then use the same patter for the new word. If there are more options, checking frequency of all generated word forms will show us what is the most probable one.

For example, the word "dovolatel" ends with known Czech suffix *-tel* (the suffix expresses that the derived noun denotates the subject of the action of the base verb [8]) and therefore it is very likely that it will be declined in the same manner as other words with this suffix (krotitel, podnikatel, ...).

A manual check is needed at the moment, though.

3 Desam

DESAM [6] is a disambiguated corpus of Czech texts originating from newspapers and scientific magazines. A lemma and morphological tag is specified for each token in the corpus. However, although manually disambiguated, the DESAM corpus is far from perfect. Some tokens were originally not assigned a tag, so the noun tag `k1` was added to all of them (most of those words are proper nouns that are unknown to the morphological analyzer; however, there are exception such as typos, rare words that the morphological analyzer does not contain or words written in non-standard manner). In the original version, there were 16,697 tokens with this tag and no further specification (1.7 % of all tokens).

Further on, even words that are properly disambiguated do not always match the lexicon entries available in Majka. This is because later changes in Majka were never properly entered in DESAM. Modifications in Majka mostly mean deleting one of possible POS interpretations for words where linguistic agreement isn't high enough and therefore annotation wasn't consistent.

The discrepancy between the current version of Majka and DESAM was solved semi-automatically by these steps:

1. If there is only one tag offered by the current version of Majka, it is considered correct. For example, for the word *sotva* (barely), three different POS tags were used in DESAM (adverb, a conjunction and particle). As the distinction isn't clear, the only available POS tag is now adverb, and all occurrences in DESAM were changed to adverb.
2. If more than one possible tag is offered in Majka, a table was created listing the options. In some cases, I could delete the ones that were clearly incorrect in the specific context and it wasn't necessary to check each case. For example, the word už (already) was tagged as particle (`k9`) in the original corpus. Now, the choice is between three tags: už (`k6eAd1tT`), už (`k5eAaImRp2nSrD`), už (`k5eAaPmRp2nS`), but the two later actually refer to lemma *úžit* (to make something narrow). In this case, we can choose the adverb tag without a manual check.
3. All other tokens found in Majka (but with multiple possible tags) were annotated manually.
4. For out-of-vocabulary tokens, the tag `k1` (noun) stayed in place.

After the aforementioned changes, 15,669 tokens were left with `k1` (noun) tag[2].

[2] This number should not be compared with the previously stated number of tokens with this tag (16,697), because MWEs were not taken into account. As described in the next section, 3,593 new tokens submerged by splitting MWEs. Altogether, we have therefore decreased the number of tokens with tag `k1` from 20,290 to 15,669.

To provide a better idea about the main changes that were performed in the corpus, Table 2 shows the tokens that were changed the most often in DESAM. For example, the token *i* (also) was originally tagged mostly as particle (in 70 %) and as conjunction in the remaining 30 %. While the particle interpretations is possible, it is very rare and by no means should cover 70 % of the cases. It is also not a straightforward task to distinguish those categories, so the only possible tag for *i* is conjunction now.

Table 2: The most frequently changed tokens in DESAM.

frequency	word	original tag	new tag
508	ještě	k9	k6eAd1
575	a	k9	k8xC
579	tedy	k9	k8xC
625	totiž	k9	k8xC
654	proto	k8xC	k6eAd1xC
686	tak	k9	k6eAd1tMtQxCxD
705	jako	k9	k8xS
778	jak	k8xS	k6eAd1yR
822	až	k9	k8xS
1,225	také	k9	k6eAd1
2,203	však	k9	k8xC
3,662	i	k9	k8xC

3.1 Multiword expressions

In the original data, some multi-word expressions were treated as one token. While justified in some cases, the annotation was not consistent and the same expressions sometimes appeared as one token and other times they were separated into multiple tokens. Also, this is not a desired state, as the current tools we are using would never recognize such MWEs as a single token.

To put it into numbers, there were 2,733 MWEs in the corpus. Once separated by spaces, it would make 6,326 tokens. Most MWEs were proper nouns (names of people or organizations), but there were others such as: *Na rozdíl od, Z hlediska, Mimo jiné, V důsledku, Ve srovnání s, V souvislosti s, V porovnání s*.

3.2 The results of Desam & Majka unification

As a simple mean to measure the effect of the modifications, `RFTagger`[3] was trained with each version of DESAM. A comparison of accuracy achieved for each version is provided in Table 3. The last column depicts results of `RFTagger` after various modifications described in [5]. In short, the work consists in altering tagset by:

- adding new attributes to specific words/group of words that linguistically differ from the rest, but the information is not considered in the current tagset (e.g. proper nouns or the word *být* that is mostly used as auxiliary verb in Czech texts).
- deleting attributes, but in the way that would not delete any information that cannot be obtained by the lexicon (e.g. change order of the attributes, delete the information about degree of adjectives and adverbs, that would be added back after training from the lexicon).

Table 3: Error rates of `RFTagger` trained on three different versions of DESAM. We consider the number "error `kgncp`" as the most important, as only attributes that are linguistically well-defined and cannot be simply taken from a dictionary are taken into account.

	original	original %	Majka	Majka %	changed	changed %
error	84,385	8.616	73,077	7.462	67,066	6.848
error in kgncp	**81,078**	**8.279**	**69,710**	**7.118**	**61,632**	**6.293**
k	20,775	2.121	9,634	0.984	7,696	0.786
c	38,058	3.886	37,976	3.878	32,414	3.31
g	19,500	1.991	19,284	1.969	18,859	1.926
n	11,547	1.179	11,013	1.125	10,366	1.058
p	15	0.002	12	0.001	13	0.001

Table 3 shows that the difference in the error rate of `RFTagger` trained on the original data and the version after matching DESAM to the current version of Majka is predominantly in the POS attribute, which is not surprising, as this is mostly what was altered in this step. The last version shows slightly better numbers in all categories, the biggest difference being the case.

The error rate of the tagger used in the new pipeline is therefore 6.293 % when being measured on the `kgncp` attributes (i.e. part-of-speech, grammatical case, number, gender and person) only.

[3] This tagger was chosen for simplicity reasons, as it is easy to use and very fast, making it possible to do various experiments in limited time span and evaluate them using the 10-fold cross-evaluation.

4 Known issues & future work

- Lemma disambiguation – lemmas in Czech language are rarely ambiguous when POS tag is taken into account. However, if such case occurs (for example for word form *karet*, there are two possible lemmas – karta (card) or kareta (loggerhead sea turtle), there is currently no disambiguation performed.
- Guesser for unknown words – the suggested solution for incorrectly guessed lemmas was drafted a long time ago (see [4]), but so far it was not implemented into the pipeline.

5 Conclusions

In this paper, we described a new version of the Czech pipeline used in Sketch Engine for corpus processing. The new pipeline uses a tagset that follows the most recent version of the Majka morphological database and uses the `RFTagger` trained on a compatible corpus (DESAM) for disambiguation. Previous experiments evaluating the `RFTagger` showed an accuracy of 93.7 % when measured on the `kgncp` attributes only.

Acknowledgements. This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2023062.

References

1. Jakubíček, M.: Rule-Based Parsing of Morphologically Rich Languages [online]. Disertační práce, Masarykova univerzita, Fakulta informatiky, Brno (2017 [cit 2024-11-17]), `https://is.muni.cz/th/h1xfz/`, sUPERVISOR : Aleš Horák
2. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen Corpus Family. 7th International Corpus Linguistics Conference CL 2013 (07 2013)
3. Kovařík, F., Kovář, V., Blahuš, M.: On Rapid Annotation of Czech Headwords: Analysing the First Tasks of Czech Dictionary Express. In: Lexicography and Semantics, Proceedings of the XXI EURALEX International Congress (2024)
4. Kovář, V., Jakubíček, M.: DMoG: A Data-Based Morphological Guesser. In: RASLAN (2018)
5. Ohlídalová, V.: Improvements of the tagset used for automatic morphological analysis of Czech [online]. Master's thesis, Masaryk University, Faculty of ArtsBrno (2023 [cit 2024-04-24]), `https://theses.cz/id/hftnho/`, supervisor: RNDr. Miloš Jakubíček, Ph.D.
6. Pala, K., Rychlý, P., Smrz, P.: DESAM - Annotated Corpus for Czech. In: Conference on Current Trends in Theory and Practice of Informatics (1997)
7. Schmid, H., Laws, F.: Estimation of Conditional Probabilities With Decision Trees and an Application to Fine-Grained POS Tagging. In: Scott, D., Uszkoreit, H. (eds.) Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). pp. 777–784. Coling 2008 Organizing Committee, Manchester, UK (Aug 2008), `https://aclanthology.org/C08-1098`

8. Šimandl, J.: Slovník afixů užívaných v češtině. Karolinum (2016)
9. Šmerk, P.: Unsupervised Learning of Rules for Morphological Disambiguation. Lecture Notes in Computer Science **3206** (2004)
10. Šmerk, P.: Fast Morphological Analysis of Czech. In: RASLAN (2009), `https://api.semanticscholar.org/CorpusID:3550809`

Rubric Extraction for Popular Science Articles in the Russian Language

Maria Khokhlova
and Natalia Safonova

St Petersburg State University, Universitetskaya emb. 7-9-11,
199034 St Petersburg, Russia
m.khokhlova@spbu.ru, st095325@student.spbu.ru

Abstract. The paper deals with topic analysis of a corpus compiled from popular science articles in the Russian language. The paper describes the procedures of corpus construction, data preprocessing, and model training on its basis. The purpose of our study is to compare the effectiveness of different methods in identifying topic words and specifying rubrics for the texts. We dwell on topic modeling using Latent Dirichlet Allocation model. The corpus was built on texts from the journal "Nauka i Zhizn" ("Science and Life"), containing articles of the news section from 2015 to 2024. We analyzed the topic diversity of the texts in the rubric and compared the topic words extracted automatically with the tags that were manually attributed by the authors of the articles. The results show that the algorithm can be used to analyze the content of texts as well as enable a more effective information retrieval.

Keywords: Natural language processing, Machine learning, Topic modeling, Latent Dirichlet Allocation

1 Introduction

Topic modeling is an effective tool for processing textual data, which allows combining documents that are close in their content. The method is widely used in as computational linguistics [1], sociology [2], bioinformatics [3] and other fields. Its results can be used in information retrieval, text structure analysis, automatic information extraction, as well as facilitates the analysis of large amounts of data and contributes to the improvement of models related to artificial intelligence.

In the study we evaluate the effectiveness of the probabilistic topic model LDA [4] applied on popular science texts in Russian. The results allowed us to analyze the diversity of the collected corpus, and also to compare the automatically identified words (and hence assigned topics) with the hashtags manually attributed by the authors of the articles.

The paper has the following structure. The Introduction explains the motivation of the study. Section 2 gives an overview of the approaches. The next

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 109–118, 2024.

section describes the design of the corpus compiled for the experiment. Section 4 discusses the results of the experiment. The last section concludes the paper and proposes plans for future work.

2 Topic modelling approaches

One of the crucial tasks in analyzing large amounts of data is to discover some common characteristics among the units within the collection. When it concerns textual data analysis, that task can often involve identifying the range of topics and concepts that appear in a document. It is not difficult for a human to determine what the text is about, but special procedures are required for automatic definition of topics by automatic systems.

Studies devoted to automatic natural language processing have demonstrated that topic modeling can enhance the performance in sentiment analysis [5], as well as automatic abstracting and summary generation [6]. The results of topic modeling techniques have potential for use in the analysis of large collections of documents, since they allow the extraction of the main information relevant for text comprehension. Originally, topic models were offered only as a tool for information extraction and document annotation [7], but at present, the possibilities of their use have expanded considerably. In addition to their wide range of applications in different scientific fields, topic modeling has a significant role for computer vision [8] and recommendation systems.

A topic model is a type of a statistical model, a useful tool for discovering the hidden semantic structure in a set of documents, namely a set of topic words [9]. The building of a topic model is based on the assumption that a text is a randomly selected collection of independent words ("bag of words"), which is produced by certain topics. In this respect, topic modeling refers to the restoration of the probability distributions of all the topics in the text. The generated topic model calculates the degree to which a document belongs to each topic, and also measures the level of accuracy of specific terms in representing a given topic. The model is based on fuzzy biclustering meaning that words and documents are simultaneously clustered into the same "topic clusters". Fuzzy clustering assumes that one word can be assigned with different probabilities to several clusters and, alternatively, one document can be assigned to several clusters with different probabilities [10]. The number of extracted topics can either be chosen by the researcher using the empirical data, or can be calculated by comparing the models automatically. To select the best model, calculating a coherence measure reflecting the level of semantic similarity between words in the same topic can be helpful. Perplexity, which shows how well the model predicts new data, is also an important characteristic for evaluating model performance.

As mentioned above, topic models have been used as information retrieval tools and for automatic indexing of documents in data mining. The models allowed solving the problem of inaccurate document retrieval. Topic models first appeared in the 1990s, as an increase in the processing power of computers

and the transition to machine learning enabled the use of statistical approaches for natural language processing. Among the first topic models introduced, one well-known model is LSI (Latent Semantic Indexing) [11]. Another model, pLSA (Probabilistic Latent Semantic Analysis) was proposed in 1999 [12]. Both models were based on the ideas of distributive semantics, which determines the degree of semantic proximity between text units by designing a vector space. The basic principle of these models is to map documents and terms into a representation in a latent semantic space.

Active development of topic models took place in the 2000s, when new efficient algorithms such as LDA (Latent Dirichlet Allocation) and NMF (Non-Negative Matrix Factorisation) were introduced by groups of researchers [13]. LDA is recognised as the most advanced and efficient for processing large text collections [14].

The models mentioned above can be classified into algebraic and probabilistic models. Recently, models that combine a probabilistic approach with the use of a distributed vector model have emerged. These include LDA2Vec, Top2Vec, and BERTtopic. Naturally, such models have an advantage over algorithms that represent the corpus as a bag-of-words. They reduce the losses associated with this representation, since the quality of semantic text structure representation they provide is higher [15].

Mainly, studies using topic models are carried out on the material of scientific texts, as well as on the texts from social media. For instance, algorithms are used to investigate the evolution of topic structure of journals [16] or to analyze the development of a particular topic within a single magazine [17]. Thematic structure studies are also conducted on the basis of fiction and poetic texts [18]. Another study has demonstrated that, in combination with other methods, the topics that are extracted using LDA can be used to predict the genre or subgenre of a work of literature [19].

3 Corpus construction

As the material for our study we used news articles published on the website "Nauka i Zhizn" ("Science and Life"). These texts are characterized by thematic diversity, while some of the materials are tagged with topic labels (which can be regarded as manually assigned rubrics). We aimed to examine the thematic range of articles, as well as to organize thematically similar texts into an appropriate number of groups. It is worth noting that the journal's website offers a search system, but this system does not provide an opportunity to select materials on a particular theme or field of science, and this fact additionally emphasizes the relevance and practical significance of our work. The search is supposed to be carried out by keywords, which the user has to define on their own, and this may entail certain difficulties. The result is a list of materials sorted by their relevance: the articles in which the key expression occurs most frequently are regarded as more relevant.

However, keywords, marking the topic of the article, do not always have to be the most frequent, and in many cases the user may not get the most relevant article on the output page. This is due to the fact that news items from October 2018 only, that is only 32% of all publications in the rubric, are tagged with topic tags. We arranged the authors' tags in a data frame in order to compare them with the automatically extracted topic words from the model after the results were collected.

The corpus of popular science articles was constructed automatically with the help of the BeautifulSoup library [20]. It included 4,205 materials in Russian in the period 2015-2024 from the "News" section. The corpus had to be preprocessed in the course of the work. This process included tokenization, lemmatization, as well as removal of stop words, punctuation marks and other non-textual symbols. To perform these tasks, we used NLTK library [21] and PyMorphy3 [22]. The corpus had a total of 1,078,538 word instances after processing. The resulting data was manually verified and errors in the lemmatization were corrected where possible. The preprocessing procedure was necessary because LDA works following the bag-of-words principle, which suggests that the model does not take into account grammatical and syntactic criteria, but rather relies on the frequency of the word occurrence in the document.

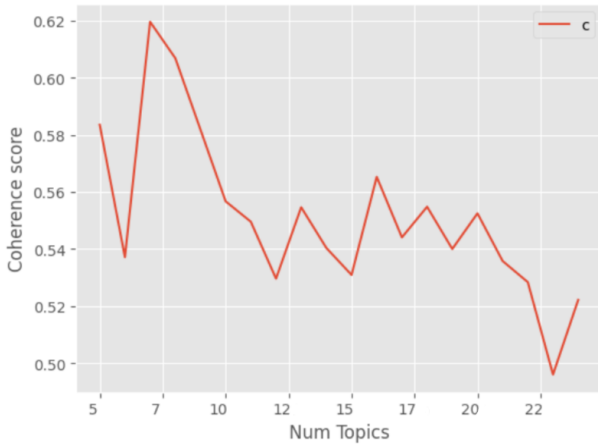
Furthermore, the corpus had to be additionally filtered in the course of the work, since some words with high frequency were assessed as irrelevant for the topic in the model we first obtained. For example, the terms *nauka* 'science' (with an absolute frequency above 10,000 in the corpus), *statya* 'article' (8,000), and *issledovatel'* 'researcher' (above 12,000) all appeared in the initial results. Such words cannot provide a meaningful representation of the topic because they are common to most texts in the corpus. We identified the most frequent words in the model dictionary that are not of interest in the topic model and excluded them from the corpus (e.g., *nauchny* 'scientific', *issledovaniye* 'research, study', *statya* 'article, paper', *universitet* 'university', *experiment* 'experiment', *resultat* 'result').

4 Results of the experiment

To train the LDA model, we used the Gensim machine learning library [23]. The number of topics was selected on the coherence measure (Figure 1.). The coherence index is the highest and reaches 0.62 specifically with 7 topics. For each selected topic a name, or label, was assigned manually. The perplexity level of the model has reached -8.57, which is a solid indicator of successful performance.

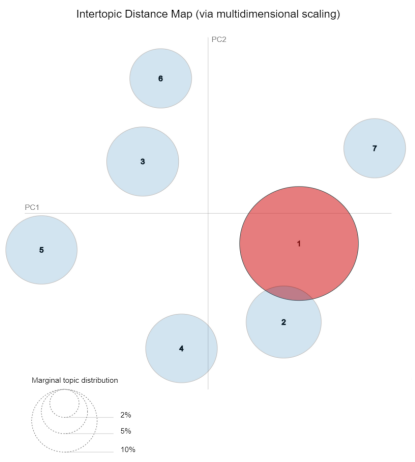
Table 1 shows the results of the LDA topic modeling. For the convenience of the presentation, we have chosen the top 15 topical words out of 30 extracted. In the process of topic labeling, we aimed to reflect the main idea that unites the words that constitute the topic. If a variety of topics were distinguished in a single theme, the name was assigned according to the main idea and in accordance with the content of the documents most typical for it. The presented

Fig. 1: Coherence



lists are vast and hence they can suggest other interpretation. Some labels reflect the topics more accurately, which is dependent on the degree of semantic homogeneity of the words within the topic. Figure 2 shows a visualization of the distances between the topics that are calculated in the system using multidimensional scaling.

Fig. 2: Coherence



We should note that the obtained topics are rather broad topics-directions, and more specific topics can be identified within them. For example, topic_1 which was determined as genetic engineering includes topic words from articles

Table 1: Result of topic modeling with LDA.

#	Label	Topic words (Russian)	Translation
Topic_1	Genetic Engineering	*kletka*, *gen*, *belok*, *dnk* *bolezn'*, *bakteriya*, *immunnyy*, *mysh*, *virus*, *molekula*, *tkan'*, *sistema*, *mutaciya*, *kletochnyy*	'cell', 'gene', 'protein', 'dna', 'disease', 'bacterium', 'immune', 'mouse', 'virus', 'molecule', 'tissue', 'system' 'mutation', 'cellular'
Topic_2	Language Psychology	*socialnyy*, *rebyonok*, *spat*, *vyigrysh*, *igra*, *yazyk*, *uchastnik*, *povedeniye*, *vliyat*, *rech*, *svyaz*, *kontekst*, *kotik*	'social', 'child', 'to sleep', 'win', 'game', 'language', 'participant', 'behaviour', 'to influence', 'speech', 'connection', 'context', 'kitty'
Topic_3	Astrophysics	*zvezda*, *sistema*, *chyornyy*, *planeta*, *temperatura*, *kosmicheskiy*, *atmosfera*, *gaz*, *obyekt*, *massa*, *dyra*, *solnechnyy*	'star', 'system', 'black', 'planet', 'temperature', 'cosmic', 'atmosphere', 'gas', 'object', 'mass', 'hole', 'solar'
Topic_4	Evolution	*derevo*, *rastenie*, *zhivotnoye*, *samec*, *samka*, *ryba*, *ptica*, *zver'*, *dinozavr*, *rasti*, *bolshiy*, *gruppa*, *telo*, *pochva*, *morskoy*	'tree', 'plant', 'animal', 'male', 'female', 'fish', 'bird', 'beast', 'dinosaur', 'to grow', 'larger', 'group', 'body', 'soil', 'marine'
Topic_5	Archaeology	*zemletryasenie*, *drevnij*, *region*, *kost'*, *kolco*, *kamen*, *nahodka*, *tysyacha*, *arheolog*, *naslat*, *godichnyj*, *peshchera*, *sovremennyj*	'earthquake', 'ancient', 'region', 'bone', 'ring', 'stone', 'find', 'thousand', 'archaeologist', 'to lay', 'year', 'cave', 'modern'
Topic_6	Molecular Biology	*ispolzovat*, *molekula*, *himicheskiy*, *veshchestvo*, *material*, *kvantovyj*, *svojstvo*, *pomoshch*, *metod*, *struktura*	'use', 'molecule', 'chemical', 'substance', 'material', 'quantum', 'property', 'help', 'method', 'structure'
Topic_7	Neurobiology	*mozg*, *neyron*, *signal*, *mysh*, *aktivnost'*, *son*, *sistema*, *dofamin*, *nervnyy*, *ritm*, *nejronnyy*, *pamyat*, *glimfaticheskiy*	'brain', 'neuron', 'signal', 'mouse', 'activity', 'sleep', 'system', 'dopamine', 'nerve', 'rhythm', 'neural', 'memory', 'glymphatic'

tagged as “medicine”, “molecular biology”, “methodology of science”, and “cognitive psychology”. This indicates that although the texts can be grouped into one large domain, they are not qualitatively homogeneous. For example, within one topic, at least 10 subtopics can be identified, which suggests a great thematic diversity of news materials.

The same applies, for example, to topic_5, the topic words of which were selected from texts in the fields of anthropology, human evolution, and archaeology. A particular interest is topic_2, which initially raised difficulties in assigning a marker. In this case topic words that represent the names of subfields within one broad topic were assigned to one topic. This topic includes texts about language acquisition by children, learning a foreign language, the brain and language, and also how pets react to speech signals. From these latter texts, the terms “kitty” and “dog” (from the top 30) were among the extracted words. Therefore, the model captures the most general topics that could serve as sections on the website and would also contain some subsections.

The distance visualization also reveals that the model has clustered the texts into seven distinct groups, which, with the exception of the topics 1 and 2 which are similar in some respects, do not overlap and are at a remote distance from each other.

It is also necessary to dwell on the comparison of the author’s tags and the results obtained. As an example, we present a few article titles and their tags from the collected dataset (Table 2).

We can observe that while hashtags carry valuable information about the thematic content of the article, in some cases they are assigned rather arbitrarily (both broadly and narrowly defined topics are observed). For this reason, it is difficult to make an automatic comparison between them and the topics identified in the course of our work. The tags that could most fully and generally relate the text to a particular field of scientific knowledge are not always attributed by the authors (Table 2). The text could have been attributed to the field of biology, but the authors do not assign a more generalizing tag.

Also, in spite of our careful attention to the preprocessing stage, some errors in the output could not be avoided. For example, the top 30 in topic_4 contains the term *nibyt*, which is not really a word, but a negation and a verb (‘notbe’). Similarly, the comparative degree of an adjective, which should have been converted to the initial form bolshiy (‘bigger’), appeared in topic_4. These observations show that further error checking of the corpus is necessary to obtain better results in the future.

5 Conclusion

In the paper, we considered an experiment of clustering a text collection of popular science news articles by rubrics. A probabilistic topic-based LDA model was used, which is a tool that discovers the distribution of topics across documents and the distribution of topic words across topics. The paper evaluated the ef-

Table 2: Result of topic modeling with LDA.

#	Title	Tags
1	*Samki lyubyat umnyh samcov* 'Females like smart males'	*#povedenie zhivotnyh*, *#brachnoye povedeniye*, *#intellekt zhivotnyh*, *#evolyuciya* 'animal behaviour', 'mating behaviour', 'animal intelligence', 'evolution'
2	*Muzhchiny i zhenshchiny pomnyat bol' po-raznomu* 'Men and women remember pain differently'	*#nejrobiologiya*, *#gormony*, #sensornye sistemy 'neurobiology', 'hormones', 'sensory systems'
3	*Ryby ne lyubyat staryh znakomyh* 'Pisces don't like old acquaintances'	*#povedenie zhivotnyh*, *#ryby* 'animal behaviour' 'fish'
4	*Nejrony mozga rassmotreli po sloyam* 'The neurons of the brain have been examined layer by layer'	*#nejrobiologiya*, *#nejrony*, *#mozg*, *#kora mozga*, *#mikroskopiya* 'neurobiology', 'neurons', 'brain' 'brain cortex', 'microscopy'

fectiveness of the model and the obtained results were compared with the tags manually attributed by the authors of the texts.

The experiment demonstrated the high efficiency of the LDA statistical model applied to the corpus of popular science articles in Russian. In future, the preprocessing stage will involve additional corpus filtering at in order to eliminate possible lemmatization errors. The corpus can be supplied with older materials to study the dynamics of the journal's thematic diversity.

Acknowledgements. The work by Maria Khokhlova in the presented research was supported by the Russian Science Foundation, project No. 24-28-00937, `https://rscf.ru/en/project/24-28-00937/`.

References

1. Kirina, M.A.: Comparison of thematic models based on LDA, STM and NMF for qualitative analysis of Russian fiction prose of small form. NSU Vestnik, Linguistics and Intercultural Communication 2(20), 93–109 (2022)

2. McFarland, D.A., Ramage, D., Chuang, J., Heer, J., Manning, C.D., Jurafsky, D.: Differentiating language usage through topic models. Poetics 41, 607–625 (2013)
3. Liu, L., Tang, L., Dong, W., Yao, S., Zhou, W.: An Overview of Topic Modeling and Its Current Applications in Bioinformatics. SpringerPlus, 5, 1608 (2016)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. The Journal of Machine Learning Research, vol. 3, pp. 993–1022 (2003)
5. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375–384 (2009)
6. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 19–25 (2001)
7. Deerwester, S., Dumais, S.T., Furnas G.W., Landauer. T. K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science, vol. 41 (6), pp. 391–407 (1990)
8. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 524–531. San Diego, CA, USA (2005)
9. Blei, D.M.: Probabilistic topic models. Communications of the ACM, 55(4), 77–84 (2012)
10. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. Journal of Machine Learning Research, vol. 14(40), pp. 1303–1347 (2013)
11. Papadimitriou Ch., Tamaki, H., Raghavan, P., Vempala, S.: Latent semantic indexing: a probabilistic analysis. In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS '98), pp. 159–168. Association for Computing Machinery, New York, NY, USA (1998)
12. Hofmann, T.: Probabilistic Latent Semantic Analysis. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 289–296 (1999)
13. Lee, D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature (401), 788–791 (1999)
14. Blei, D.M., Lafferty, J. D.: Topic models. Text mining, pp. 101–124. Chapman and Hall/CR (2009)
15. 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, vol. 1, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics (2019)
16. Odden, T., Marin, A., Rudolph, J. L.: How has Science Education Changed over the last 100 years? An analysis using natural language processing. Science Education (105), 653–680 (2021)
17. Jacobi, C., Van Atteveldt, W., Welbers, K.: Quantitative analysis of large amounts of journalistic texts using topic modelling. In: Rethinking Research Methods in an Age of Digital Journalism, pp. 89–106. Routledge (2018)
18. Rhody, L.M.: Topic Modelling and Figurative Language. Journal of Digital Humanities, 19–35 (2012)
19. Schöch, C.: Topic modeling genre: an exploration of French classical and enlightenment drama. Digital Humanities Quarterly, vol. 11(2) (2017). `https://www.digitalhumanities.org/dhq/vol/11/2/000291/000291.html` Last accessed 1 Nov 2024.
20. BeautifulSoup library, `https://www.crummy.com/software/BeautifulSoup`. Last accessed 1 Nov 2024.

21. NLTK library, https://www.nltk.org. Last accessed 1 Nov 2024.
22. PyMorphy3 library, https://pypi.org/project/pymorphy3. Last accessed 1 Nov 2024.
23. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50 (2010). https://is.muni.cz/publication/884893/en.

Part IV

NLP Applications

Lexical Density in Slovak Speech: A Non-invasive Indicator for Alzheimer's Disease and Mild Cognitive Impairment

Nataliia Časnochova Zozuk, Lívia Kelebercová, and Daša Munková

Constantine the Philosopher University in Nitra, Tr. A. Hlinku 1, SK, 949 01 Nitra, Slovakia
nataliia.casnochova.zozuk@ukf.sk
lkelebercova@ukf.sk
dmunkova@ukf.sk

Abstract. Background: Alzheimer's disease (AD) and mild cognitive impairment (MCI) are increasingly prevalent neurodegenerative conditions that impact cognitive and linguistic functions. Early detection is crucial for timely intervention, yet traditional diagnostic methods, such as neuroimaging and cerebrospinal fluid analysis, are invasive and costly. This study investigates whether linguistic markers, specifically lexical density in spoken language, can serve as reliable, noninvasive indicators of cognitive impairment. **Methods:** Using data from 214 participants (healthy (CN), MCI, and AD) collected via a picture description task, we applied natural language processing (NLP) techniques to analyze various measures lexical density (*V1/W*, *R2*, *ADJ/W*, *ADV/W*). **Results:** Statistical analysis revealed significant associations between certain lexical metrics and cognitive impairment levels. Specifically, measures such as *V1/W* and *H1* differentiated healthy participants from those with MCI, while ADJ/W and ADV/W were particularly effective in distinguishing AD from cognitively normal participants. **Conclusion:** These findings suggest that linguistic features, related to lexical density, can provide insights into cognitive health and may offer a valuable tool for early detection of AD and MCI. Future research should explore broader linguistic metrics and additional language tasks to enhance diagnostic accuracy and facilitate the development of automated, speech-based screening tools.

Keywords: Lexical density, Mild cognitive impairment, Alzheimer's disease.

1 Introduction

Alzheimer's disease (AD) is the most common form of dementia and is expected to affect an increasing number of people each year as life expectancy rises [1-2]. Characterized by irreversible neuronal loss, particularly in the cortex and hippocampus, AD results in progressive memory impairment (or loss) and behavioral changes and generally follows a preliminary stage known as mild cog-

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 121–128, 2024.

nitive impairment (MCI) before advancing to full dementia [3-4]. Early detection of cognitive decline associated with MCI and AD is critical, as it can help reduce healthcare costs and lessen the emotional impact on both individuals with AD and MCI and caregivers [5]. While current diagnostic techniques often rely on invasive procedures, such as magnetic resonance imaging (MRI) or cerebrospinal fluid (CSF) analysis, these methods are costly and resource-intensive.. However, AD is associated with specific impairments (alternations) in continuous speech, non-invasive speech and text analysis via artificial intelligence offers a promising and cost-effective diagnostic tool. From a clinical perspective AD is typically characterized not only by structural changes in the brain and the presence of certain proteins but also by alterations in coherent (spoken) speech [6]. Speech in people with AD is often described as "flat," exhibiting reduced lexical diversity, simplified syntax, and frequent repetition [7-8]. Since natural aging also affects speech, it is essential to distinguish age-related language changes from those indicating neurodegenerative diseases , taking into account demographic factors such as age, gender, and education [9-11]. T his study aims to identify lexical predictors for detecting AD and MCI by analyzing data from 214 participants with AD, MCI, or healthy people (CN) status through a picture description task. Linguistic features we extracted from participants´ speech samples and analyzed to compare patterns across these diagnostic groups. Following a review of related work, we present the methodology, data analysis, and results, concluding with interpretations and future implications.

2 Related Work

Numerous studies have identified specific linguistic markers that can effectively distinguish between healthy elder people and individuals with MCI or AD [8, 12-15]. One important linguistic marker associated with AD is lexical diversity, which tends to be lower among individuals with the disease, often resulting in reduced ability to communicate effectively [16]. Lexical diversity is defined as the ratio of unique words to the total number of words in a text or speech sample, with a higher ratio indicating greater range of vocabulary [17].

In addition to lexical diversity, lexical density is frequently examined to further understand language characteristics. Lexical density is based on vocabulary, divided into two categories – lexical (or content) words and function (or-grammatical) words. Lexical words encompass adjectives, adverbs, nouns and verbs, as they primarly convey the meaning of the text. In contrast, function words, such as pronouns, conjunctions, and prepositions, which serve a grammatical purpose rather than contributing to the primary meaning [18]. The author [15] conducted a detailed analysis of the impact of AD on selected lexical features representing various aspects of the lexicon, such as diversity, density, and sophistication (Type Token Ratio, Uber Index, Entropy, etc.). The author aimed to identify the optimal set of features and the most effective input length to maximize the classification model´s accuracy. The findings revealed that, although diversity-related metrics constituted approximately half of the

top 50 features and density-related and specificity metrics despite making up less than 10%, demonstrated superior performance, yielding higher F1 scores than diversity. Authors [19] explored the linguistic characteristics associated with AD by examining differences in the usage of lexical words (nouns, verbs, and pronouns) between individuals with AD and healthy people (cognitively normal, CN) . Their results indicated that individuals with AD used significantly fewer substantive changes than healthy (CN) people ($p < 0.01$). Additionally, they observed, that people with AD exhibited reduced lexical diversity then CN, showing an increased reliance on pronouns and diminished diversity in noun usage.

The above mention studies suggest that lexical markers may serve as effective indicators of neurodegenerative disease symptoms. Although this topic has received substantial attention in recent years [4-8], none of them has focused on an inflected language, such as Slovak.

3 Methodology

A common research method for assessing language deficits in individuals with AD or MCI involves the use of picture description tasks that feature multiple topics [4, 8, 13, 20]. In these studies, researchers applied natural language processing (NLP) techniques to extract specific linguistic features from the speech of both healthy (CN). individuals and those suffering from AD and MCI By identifying differences in these features, they trained a model - a classifier - to categorize individuals as either healthy (CN) or people with neurogenerative diseases. To control for variations in speech that may arise from factors other than disease, researchers typically analyze speech outputs using balanced dataset based on three demographic characteristics (age, gender and education) [21-23]. Considering these factors, we selected transcriptions from picture description tasks completed by both healthy participants (CN) and individuals with early or progressing neurodegenerative disease for our research.

3.1 Participants

The data for this study was sourced from the Early Warning of Alzheimer (EWA) project [24]. Speech samples, recorded through a picture description task, included both healthy individuals and patients diagnosed with cognitive impairments. Participants met criteria such as being over age 50 and free from serious psychiatric or neurological conditions, speech disorders, or severe visual impairment. Healthy individuals scored 24 or higher on the MoCA cognitive function test, while patient participants had a score of at least 18.

Voice recordings were transcribed using an Automatic Speech Recognition (ASR) tool developed by the Slovak Academy of Sciences, followed by manual review. This study analyzes responses from 107 cognitively healthy individuals, 63 with Alzheimer's disease, and 44 with MCI, representing a balanced subset from the larger EWA dataset of around 1,614 participants.

3.2 Metrics

We used a suitable tool from one of the libraries of the Python programming language for the texts obtained from the language expressions of the probands of the individual determined groups. It was a Stanza library for tokenization, lemmatization, and other natural language processing tasks to apply text complexity measures. Subsequently, we automatically extracted 10 linguistic signs (characteristics) from the texts, focusing mainly on lexical density: *V1/W* (*Words that occur only once / Total Words*), *V2/W* (*Words that occur twice / Total number of words*), *H1* (*Hapax diversity measure*), *R1* (*Lexical richness measure, with hapax legomena*), *R2* (*Lexical concentracion measure, with repeated words*), *NOUN/W* (*Nouns / Total Words*), *ADJ/W* (*Adjectives / Total Words*), *ADV/W* (*Adverbs / Total Words*), *VERB/W* (*Verbs / Total Words*).

Metrics *H1, H2, R1* and *R2* were calculated using formullas (1-4):

$$H1 = \frac{\log(T) \cdot 100}{1 - V1/T}, \tag{1}$$

$$H2 = \frac{\log(T) \cdot 100}{1 - V2/T}, \tag{2}$$

$$R1 = \frac{\log(N) \cdot 100}{1 - V1/N}, \tag{3}$$

$$R2 = \frac{\log(N) \cdot 100}{1 - V2/N}, \tag{4}$$

where T – number of unique words, $V1$ – number of words that occur only once, $V2$ – number of words that occur twice, N – number of total words.

3.3 Research Assumptions

The presented article traces the relationship between individual measures of lexical density determined from transcriptions of audio recordings of speech during the picture description task and the level of cognitive impairment of the patients (dgn), while this level was marked by three categorical values, namely (0 - CN,healthy patient, 1 - MCI, and 2 - AD). As part of the experiment, the following assumptions were made:

- Patients with mild cognitive impairment describe situations with lower lexical density (measured by metrics *V1/W, V2/W, H1, H2, R1, R2, NOUN/W, ADJ/W, ADV/W, VERB/W*) compared to healthy patients due to mild difficulties with expressing and processing information.
- Patients with Alzheimer's disease describe a situation with even lower lexical density (measured by metrics *V1/W, V2/W, H1, H2, R1, R2, NOUN/W, ADJ/W, ADV/W, VERB/W*) compared to patients with mild cognitive impairment due to more serious language deficits and cognitive limitations.

4 Results

The null hypothesis (H0) can be formulated as follows: There is no statistically significant relationship between the selected indicators of lexical density and the degree of cognitive disability of the patient.

Due to non-normal data distribution, as indicated by the Shapiro-Wilk test ($p < 0.05$), non-parametric methods were used to assess associations between lexical density measures (*V1/W, V2/W, H1, H2, R1, R2, NOUN/W, ADJ/ W, ADV/W, VERB/W*) and the degree of cognitive impairment of the patient, non-parametric procedures were used.

Table 1 presents the degree of association between the above-mentioned measures and the degree of cognitive level of the respondent (*dgn*) in the observed 214 samples. Significant relationships were found for V*1/W, R2, ADJ/W* ($p < 0.001$), as well as for *H1, ADV/W* ($p < 0.01$).

Table 1: Gamma coefficients: lexical density x dgn

Metric	Valid N	Gamma	Z-score	p-value
V1/W & dgn	214	0.203221***	3.48637	0.000490
V2/W & dgn	214	0.053228	0.91216	0.361683
H1 & dgn	214	0.145262**	2.49158	0.012718
H2 & dgn	214	-0.028523	-0.48882	0.624968
R1 & dgn	214	-0.064127	-1.10081	0.270980
R2 & dgn	214	-0.555665***	-9.53960	0.000000
NOUN/W & dgn	214	-0.014862	-0.25463	0.799012
ADJ/W & dgn	214	-0.278981***	-4.78137	0.000002
ADV/W & dgn	214	-0.155380**	-2.66690	0.007655
VERB/W & dgn	214	0.074857	1.28342	0.199345

Note: *** - $p < 0.001$, ** - $p < 0.01$, * - $p < 0.05$

For further investigation, the assumption of homogeneity of variances was verified by Levene's test. Due to the inequality of variances ($F > 1.783, p < 0.05$), non-parametric tests were subsequently chosen for the comparison of several independent samples.

From Table 2, it can be seen that *R2* can distinguish between healthy people and people with neurodegenerative disease but cannot distinguish a specific disease. Conversely, *V1/W* and *H1* metrics can distinguish healthy patients from those with mild cognitive impairment. The ability to discriminate between healthy patients and patients with Alzheimer's disease was demonstrated by the proportion of adjectives to all words (*ADJ/W*) as well as the proportion of adverbs to all words (*ADV/W*).

Table 2: Multiple comparisons of the measure of lexical density x dgn

Metric	Valid	Sum of Ranks	Mean Rank	0	1	2
V1/W: Kruskal-Wallis test: $H(2, N = 214) = 16.67294$, $p = 0.0002$						
0	108	9915.5	91.8102		**0.000272**	0.053241
1	44	5947.0	135.1591	**0.000272**		0.306096
2	62	7142.5	115.2016	0.053241	0.306096	
H1: Kruskal-Wallis test: $H(2, N = 214) = 11.92536$, $p = 0.0026$						
0	108	10293.5	95.3102		**0.001852**	0.372791
1	44	5862.0	133.2273	**0.001852**		0.186970
2	62	6849.5	110.4758	0.372791	0.186970	
R2: Kruskal-Wallis test: $H(2, N = 214) = 68.12768$, $p = 0.0000$						
0	108	15221.0	140.9352		**0.000007**	**0.000000**
1	44	3901.5	88.6705	**0.000007**		0.098482
2	62	3882.5	62.6210	**0.000000**	0.098482	
ADV/W: Kruskal-Wallis test: $H(2, N = 214) = 7.080144$, $p = 0.0290$						
0	108	12407.5	114.8843		1.000000	**0.033711**
1	44	5025.5	114.2159	1.000000		0.138280
2	62	5572.0	89.8710	**0.033711**	0.138280	
ADJ/W: Kruskal-Wallis test: $H(2, N = 214) = 16.71594$, $p = 0.0002$						
0	108	13370.5	123.8009		0.093404	**0.000202**
1	44	4397.0	99.9318	0.093404		0.616234
2	62	5237.5	84.4758	**0.000202**	0.616234	

5 Discussion

Our study suggests that speech lexical density serves as a significant indicator of patient cognitive health. Specific measures of lexical density, such as the proportion of words that occur only once ($V1/W$), the lexical density index based on unique words ($H1$), the repetition index ($R2$), the proportion of adjectives (ADJ/W) and adverbs (ADV/W), proved to be statistically significant factors associated with cognitive impairment. These findings support the hypothesis that alterations in language production, especially regarding lexical density and part-of-speech selection, are associated with cognitive decline severity and they align with existing research that has identified specific linguistic features associated with neurodegenerative diseases, particularly AD and MCI [8,12-15, 18].

The significance of $V1/W$ and$R2$ in identifying cognitive impairment supports the findings by [4], who noted that increased reliance on basic vocabulary reflect cognitive deterioration in AD patients, particularly in narrative speech tasks.

The findings for $H1$ and ADV/W also align with research [1] indicating that individuals with MCI display lower lexical density than healthy individuals, yet higher than those with AD. This emphasizes the potential of non-invasive, speech-based assessments in early diagnostics across languages.

While Kurdi's study [14] found the ratios of adjectives, adverbs, and verbs to total words significant, our results diverged by showing noun-verb ratios

as weaker predictors of AD. Our findings do align, however, with previous research [25] that showed only adjective and adverb ratios relative to unique words significantly differed among groups.

6 Conclusion

This experiment yielded valuable insights into the relationship between speech lexical density and cognitive impairment, demonstrating that lexical density metrics correlate with varying degrees of cognitive deficit. The presented study is non-invasive, which means that the assessment of language skills takes place without the need to interfere with the patients' physical health. This approach minimizes the psychological stress of testing, allowing patients to express themselves naturally and enhancing data validity. Considering the progressive nature of cognitive diseases such as Alzheimer's disease, our findings could be the basis for the development of new diagnostic methods and interventions that could help early detection and monitoring of cognitive changes. Future research could also explore other aspects of speech complexity for a holistic understanding of language and thought depending on the level of cognitive health.

Acknowledgements. This work was supported by the University Grants Agency (UGA) no. VII/4/2024.

References

1. Williams, E., McAuliffe, M., Theys, C.: Language changes in Alzheimer's disease: A systematic review of verb processing. Brain Lang. 223, 105041 (2021). https://doi.org/10.1016/j.bandl.2021.105041
2. The World Health Organization: Dementia, https://www.who.int/news-room/fact-sheets/detail/dementia (2023, accessed 7 June 2024).
3. Nussbaum, R.L., Ellis, C.E.: Alzheimer's Disease and Parkinson's Disease. N. Engl. J. Med. 348, 1356–1364 (2003). https://doi.org/10.1056/NEJM2003ra020003
4. Fraser, K.C., Fors, K.L., Kokkinakis, D.: Multilingual word embeddings for the assessment of narrative speech in mild cognitive impairment. Comput. Speech Lang. 53, 121–139 (2019). https://doi.org/10.1016/j.csl.2018.07.005
5. Robin, J., Xu, M., Kaufman, L.D., Simpson, W.: Using Digital Speech Assessments to Detect Early Signs of Cognitive Impairment. Front. Digit. Health. 3, 749758 (2021). https://doi.org/10.3389/fdgth.2021.749758
6. Bose, A., Ahmed, S., Cheng, Y., et al.: Connected speech features in non-English speakers with Alzheimer's disease: protocol for scoping review. Syst. Rev. 13(1), 40 (2024). https://doi.org/10.1186/s13643-023-02379-y
7. Rentoumi, V., Paliouras, G., Danasi, E., et al.: Automatic detection of linguistic indicators as a means of early detection of Alzheimer's disease and of related dementias: A computational linguistics analysis. In: 8th IEEE Int. Conf. Cogn. Infocommunications (CogInfoCom). IEEE, pp. 33–38 (2017).

8. Garcia, D.L., Gollan, T.H.: 15 Different Languages, Different Linguistic Markers: Predicting Which Bilinguals will Develop Alzheimer's Disease with Spontaneous Spoken Language. J. Int. Neuropsychol. Soc. 29(s1), 226–227 (2023). https://doi.org/10.1017/S135561772300334X
9. Martínez-Nicolás, I., Llorente, T.E., Ivanova, O., Martínez-Sánchez, F., Meilán, J.J.G.: Many Changes in Speech through Aging Are Actually a Consequence of Cognitive Changes. Int. J. Environ. Res. Public Health. 19(4), 2137 (2022). https://doi.org/10.3390/ijerph19042137
10. Larsson, S.C., Traylor, M., Malik, R., et al.: Modifiable pathways in Alzheimer's disease: Mendelian randomisation analysis. BMJ 359, j5375 (2017).
11. Stern, Y.: Cognitive reserve in ageing and Alzheimer's disease. Lancet Neurol. 11(11), 1006–1012 (2012). https://doi.org/10.1016/S1474-4422(12)70191-6
12. Calzà, L., Gagliardi, G., Rossini Favretti, R., Tamburini, F.: Linguistic features and automatic classifiers for identifying mild cognitive impairment and dementia. Comput. Speech Lang. (2021). https://doi.org/10.1016/j.csl.2020.101113
13. Lindsay, H., Tröger, J., König, A.: Language Impairment in Alzheimer's Disease—Robust and Explainable Evidence for AD-Related Deterioration of Spontaneous Speech Through Multilingual Machine Learning. Front. Aging Neurosci. 13, 642033 (2021). https://doi.org/10.3389/fnagi.2021.642033
14. Kurdi, M.Z.: Automatic Identification of Alzheimer's Disease using Lexical Features extracted from Language Samples. arXiv preprint arXiv:2307.08070 (2023).
15. Kurdi, M.Z.: Automatic diagnosis of Alzheimer's disease using lexical features extracted from language samples. J. Med. Artif. Intell. 7, 13 (2024). https://doi.org/10.21037/jmai-23-104
16. Fraser, K.C., Meltzer, J.A., Rudzicz, F.: Linguistic Features Identify Alzheimer's Disease in Narrative Speech. J. Alzheimers Dis. 49(2), 407–422 (2016). https://doi.org/10.3233/JAD-150520
17. Baese-Berk, M.M., Drake, S., Foster, K., Lee, D., Staggs, C., Wright, J.M.: Lexical Diversity, Lexical Sophistication, and Predictability for Speech in Multiple Listening Conditions. Front. Psychol. 12, 661415 (2021).
18. Lissón, P., Ballier, N.: Investigating lexical progression through lexical diversity metrics in a corpus of French L3. Discours. 23 (2018).
19. Williams, E., Theys, C., McAuliffe, M.: Lexical-semantic properties of verbs and nouns used in conversation by people with Alzheimer's disease. PLoS ONE 18(8), e0288556 (2023). https://doi.org/10.1371/journal.pone.0288556
20. Cummings, L.: Describing the cookie theft picture: sources of breakdown in Alzheimer's dementia. Pragmat. Soc. 10, 153–176 (2019).
21. Carter, K.C.: Language and Cognition in Mild Alzheimer's Disease. Electron. Theses Diss. 3434 (2022). https://digitalcommons.memphis.edu/etd/3434
22. Sanz, C., et al.: Automated text-level semantic markers of Alzheimer's disease. Alzheimers Dement. Diagn. Assess. Dis. 14, e12276 (2022).
23. Eyigoz, E., Mathur, S., Santamaria, M., Cecchi, G., Naylor, M.: Linguistic markers predict onset of Alzheimer's disease. EClinicalMedicine 28, 100583 (2020).
24. Rusko, M., Sabo, R., Trnka, M., et al.: EWA-DB, Slovak Database of Speech Affected by Neurodegenerative Diseases (2023). https://doi.org/10.1101/2023.10.13.23296810
25. Casnochova Zozuk, N., Munkova, D., Kelebercova, L., Munk, M.: Relationship between language features extracted through NLP and clinically diagnosed Alzheimer's disease and mild cognitive impairment in Slovak (in press).

Improvement of Language Identification Processing Speed

Emma Bednaříková and Pavel Rychlý

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{536251, pary}@mail.muni.cz

Abstract. This paper presents the Langtok model developed for the purpose of identifying the language of each token in codemix texts, i.e. texts where words from multiple languages are used. It provides insight into the development process and evaluation. Furthermore, the paper describes the utilization of the model for the filtration of large amounts of Czech text for machine translation. The main focus of this paper is conducting a set of experiments to identify the optimal conditions for input processing with the aim of decreasing the process duration.

Keywords: Codemix Text, Processing Speed, Language Identification.

1 Introduction

The identification of the language used in a text is important in a number of contexts. The most commonly employed tools for this purpose, such as fasttext [5] [4], langdetect [8], or langid [6], are designed to detect the language of an entire sentence or paragraph. However, when analyzing codemix text, it is of particular importance to be able to determine the language of each word in a given sentence or paragraph. The model presented in this paper represents a potential solution for this task. When utilizing the model for the labeling of large amounts of data, it is vital to optimize the computation process. Therefore, a series of experiments was conducted to identify the best conditions for the processing of the inputs.

2 The Langtok Model

Langtok is a model designed to identify the language of each token in the input. Although the sentence on the input may contain words from different languages, the model should be able to distinguish them and return a list of language labels where each label corresponds to a token at the same position in the sentence.

The following example illustrates the desired behavior of the model. Due to the nature of the tokenizer used, the expected output would contain slightly more labels, as the tokenizer would probably split some of the words. This example is for demonstration purposes only.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 129–138, 2024.

Input: This is a tool určený k zjištění jazyka von jedes Wortes in einem Satz.

Output: [‘eng’, ‘eng’, ‘eng’, ‘eng’, ‘ces’, ‘ces’, ‘ces’, ‘ces’, ‘deu’, ‘deu’, ‘deu’, ‘deu’, ‘deu’, ‘deu’]

3 Development

The first part of the model-building process was to create a dataset for the training. This also included choosing the languages that the model should be able to identify. Once the dataset was created, the model was trained on it.

3.1 Language Selection

This model includes 18 languages: Arabic, Czech, Danish, German, English, French, Haitian Creole, Italian, Japanese, Lingala, Dutch, Polish, Portuguese, Russian, Slovak, Spanish, Swedish, and Ukrainian.

3.2 Building of the Training Dataset

The training dataset consists of a set of modified sentences. Each sentence contains two snippets of text in a language other than the language of the sentence. The length of the snippets varies between two to six words. These snippets were inserted in a random position in the source sentence. The sentences and snippets were split by word_tokenize function from the nltk library[1], except for sentences and snippets in Japanese. In this case a different tokenizer[7] had to be utilized, due to the word_tokenize’s incompetence to tokenize text in Japanese script.

Every such modified and tokenized sentence has a corresponding list of language labels of the same length as the sentence. Each label in the label list corresponds to a token in the same position in the list of tokens.

The raw text for the creation of the dataset was collected from Flores-200 [2]. The dataset consists of three parts: train, validate, and test. The size of the train split is ca 19250 input and output pairs, and the size of the validate and test split is in both cases ca 2150 input and output pairs. This contributes to the approximate ratio of 80:10:10.

3.3 Training

The base model used for the training was Google’s BERT multilingual base model (cased) [3]. The training parameters were the same as the ones provided in the HuggingFace tutorial on training a model for token classification tasks.[1] The batch size was set to 16, the number of epochs was 3, and the learning rate was 2e-5. The training was conducted on a GPU of the Tesla T4 type, with a total training time of 28 minutes and 28 seconds.

[1] *https : //huggingface.co/docs/transformers/tasks/token_classification*

4 Evaluation

In order to evaluate the performance of the model, a confusion matrix was constructed based on the model's results for the inputs from the test dataset. Figure 1 illustrates the results for a subset of the languages. To facilitate comprehension, the values in the matrix were converted to a percentage scale.

	ces	deu	eng	por	rus	slk	spa	ukr
ces	98,11%	0,10%	0,06%	0,06%	0,00%	1,44%	0,00%	0,01%
deu	0,00%	99,48%	0,09%	0,00%	0,00%	0,02%	0,07%	0,00%
eng	0,07%	0,00%	99,12%	0,04%	0,00%	0,18%	0,02%	0,00%
por	0,04%	0,04%	0,16%	98,42%	0,00%	0,11%	0,93%	0,00%
rus	0,00%	0,02%	0,00%	0,02%	99,48%	0,00%	0,00%	0,49%
slk	1,00%	0,01%	0,01%	0,01%	0,00%	98,56%	0,03%	0,00%
spa	0,10%	0,02%	0,07%	0,57%	0,02%	0,00%	98,87%	0,00%
ukr	0,00%	0,00%	0,00%	0,00%	0,30%	0,00%	0,00%	99,70%

Fig. 1: The evaluation results for eight of the languages

The model achieved a score of 98% or above for all languages. The highest scores the model reached with Arabic and Japanese. Given the fact that both of these languages use their unique scripts that significantly differ from the scripts of other languages within the model this is not surprising.

Another unsurprising, yet nevertheless interesting phenomenon is the fact that the language pairs in which the model is observed to produce the highest error rates are in most cases two languages from the same language family, namely Czech and Slovak, Spanish and Portuguese, or Russian and Ukrainian.

5 Practical Applications

The model can be employed in a similar manner to that of other language identification tools. The two main domains are data labeling and data filtering.

5.1 Filtering Data for Machine Translation

One of the possible practical applications of the model is the filtering of data for machine translation. To build a machine translation system, a large amount of training data is needed. As these data are frequently obtained through mechanical means with minimal supervision, there is a tendency for them to contain text that is not written in the desired language. The filtration of such data may prove beneficial in enhancing the performance of the translation system.

The advantage of the Langtok model compared to traditional language identification tools is that it can identify and subsequently filter sentences that are mostly written in one language, yet contain words of another. In order to avoid the inclusion of a codemix text in the monolingual training data, the use of a token-level language identifier represents an effective solution.

However, utilization of the Langtok model may also produce certain issues. When the model was first tested for this task on Czech data, it was observed that the model filtered sentences that did not present any problems. For each sentence, the filtering algorithm calculated the distribution of the language labels. The language with the highest number of labels was selected as the language of the sentence. In instances where the language of the sentence was not identified as Czech, said sentence got filtered. In several cases, the sentence was falsely labeled as Slovak, leading to its potential removal from the training data. Furthermore, there have been instances where the language identified for all the tokens within a given sentence was completely different from Czech or Slovak. Table 1 presents the sentences that were incorrectly selected as suitable for filtration and also the language label that was used for most of the tokens of the sentence instead.

Table 1: Sentences incorrectly selected as suitable for filtration (non-Czech) from the machine translation data.

Jamesi, já nestojím o hrdinu. (slk)	To je ono. (slk)
To je v poho, no tak. (slk)	Já ti ukážu, co je to pravý muž. (slk)
Zrzek. (pol)	No a? (por)
Hele. (dan)	Omdlela. (swe)

6 Optimizing Input Processing

Since many applications of the model often require the analysis of large amounts of data, it is important to keep the processing time per input sentence to a minimum. In this section, three different strategies that show a possible improvement in reducing the process duration are presented.

6.1 Structural Modifications in Input Processing Workflow

Firstly, it is important to keep the code that provides the inference with the model well structured in order to avoid unnecessary repetition of actions that could be simplified. Secondly, the deployment of different methods should be considered. The following paragraphs present three distinct strategies that can be employed independently.

The most straightforward approach to implementation is to have a function that loads both the model and the model's tokenizer and then processes a single

sentence. However, this would require loading the model every time results for a single sentence are computed. Therefore, this strategy represents a viable yet inherently time-consuming solution. Single model and tokenizer loading would lead to a significant decrease in the time spent obtaining the results for a set of sentences.

An additional potential improvement in accelerating the computation is enabling the model to process multiple inputs concurrently. In this approach, the weights within the model are calculated for all inputs simultaneously. The deployment of this strategy has proven to be efficient. Nevertheless, concurrent processing may introduce a decrease in speed when working with large batches of inputs. This issue is addressed in the following part of this paper.

The last technique was the utilization of graphical processing units (GPUs). These provide faster computation when working with neural networks. This method requires transferring the model and the inputs to the GPU which results in a delay to the start of the calculation. However, the sole processing occurs in a shorter period than on the CPU.

In order to obtain an accurate representation of the average duration for each case, ten test runs were conducted. The values presented in Figure 2 represent the average duration of these runs.

Each run was tested on two input files; a short one containing 20 input sentences and a longer one containing 200 sentences. As anticipated, the instances where GPU was utilized exhibited inferior performance than the same cases with GPU utilization. However, it is noteworthy, that the transfer of the model to the GPU averaged 0,835 seconds. For cases 5 and 7, and 6 and 8, the results demonstrate a roughly similar duration, as in all of these cases the model and tokenizer were always loaded only once, due to the concurrent processing of inputs. The tests were conducted on a CPU device comprising 80 processors (apollo) and the GPU utilized was of the Tesla T4 type.

case	using GPU	loading model only once	simultaneous processing	duration of processing [s]	
				short file	long file
1	0	0	0	14,421	107,867
2	1	0	0	14,493	128,564
3	0	1	0	1,886	5,892
4	1	1	0	3,905	5,018
5	0	0	1	1,582	3,670
6	1	0	1	3,241	4,591
7	0	1	1	1,616	3,848
8	1	1	1	3,152	4,566

Fig. 2: Average durations of test runs

6.2 Finding the Optimal Batch Size For Input Processing

The simultaneous processing of multiple input sentences results in increased efficiency. When computing with multiple inputs, the model requires that the inputs be of the same length; thus, all inputs are padded to the length of the longest sentence. The complexity of the calculation of the result is dependent on the input length; the longer the input, the larger the matrix that will be processed during the computation. It is therefore desirable to feed the model with the inputs in batches to prevent the computation of unnecessarilly large matrices.

To estimate the ideal batch size value a series of tests was conducted. The evaluation file consisted of 2000 sentences that were processed by the model in batches. Each batch size was tested 4 times. The average duration of these tests is shown in Table 2. The same experiment was also conducted on a GPU device. The results of these tests are in Table 3. Both experiments were conducted on a CPU device comprising 32 processors (epimetheus2 and epimetheus4) and the GPU utilized was of the Tesla T4 type.

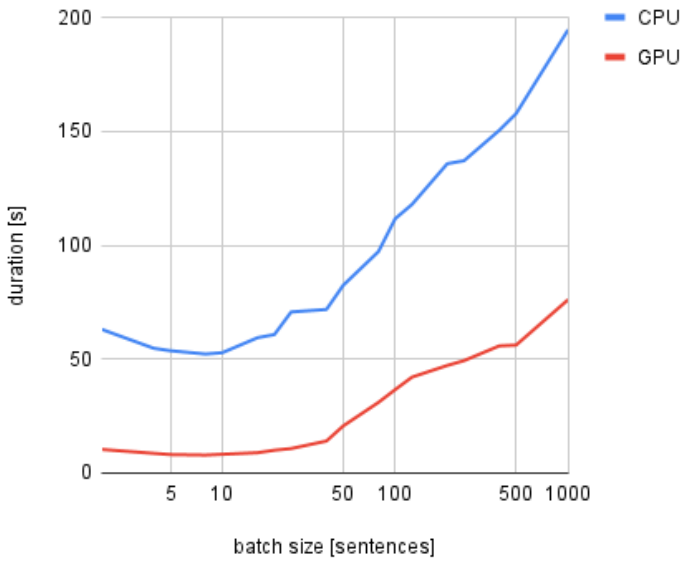


Fig. 3: Comparison of values measured on CPU and GPU

As illustrated in Figure 3, the batch sizes that offer the shortest duration for both CPU and GPU are those within the range of 4 to 10.

Table 2: Statistics from test runs on CPU

batch size [*sentence*]	overall duration [*s*]	min batch length [*token*]	avg batch length [*token*]	avg memory [*GB*]	avg %CPU
2	66,605	4	29,644	1,071	1602,000
4	57,127	5	38,246	1,150	1607,417
5	55,910	5	41,823	1,150	1607,111
8	54,458	5	49,368	1,167	1608,667
10	54,729	6	54,515	1,221	1607,526
16	62,256	7	63,896	1,265	1605,706
20	63,307	12	71,890	1,294	1604,944
40	74,205	20	91,520	1,538	1602,615
80	99,797	58	121,080	1,464	1583,880
200	138,399	119	156,900	2,511	1501,474
500	160,357	119	180,500	2,800	1443,053
1000	197,280	204	216,500	5,558	1384,526

Table 3: Statistics from test runs on GPU

batch size [*sentence*]	overall duration [*s*]	min batch length [*token*]	avg batch length [*token*]	GPU memory [*MB*]	avg %GPU
2	13,244	4	29,644	1313	52,2
4	10,605	5	38,246	1345	55,5
5	10,138	5	41,823	1361	57,2
8	10,073	5	49,368	1399	63,7
10	10,354	6	54,515	1421	69,9
16	11,022	7	63,896	1525	76,9
20	12,059	12	71,890	1571	77,1
40	15,944	20	91,520	1853	78,1
80	33,277	58	121,080	2429	77,6
200	49,428	119	156,900	3719	86,4
500	58,797	119	180,500	7333	82,4
1000	78,708	204	216,500	13373	84,7

6.3 Increasing the Number of Processes Running Simultaneously

The last strategy for optimizing the process duration was evaluating more input files concurrently. By utilizing this, efficiency can be improved, however only to a certain extent. Once the number of processes operating concurrently exceeds the available resource capacity of the device, the computational speed declines markedly as the active processes must compete for the limited available resources.

In order to ascertain the optimal number of processes that can be run simultaneously to facilitate the most efficient input processing, an experiment was conducted. The objective was to process 24 files, each of which contained 400 sentences. Initially, the files were processed one by one, and subsequently, the number of files processed simultaneously was increased. Table 4 presents the results of conducting this experiment on a CPU device comprising 32 processors (epimetheus1).

As demonstrated in Table 4, the shortest processing time was observed when two files were processed simultaneously. A mere increase of one in the number of concurrent processes resulted in a notable increase in processing time. The same task was repeated on a GPU device of type Tesla T4. The results are shown in Table 5. Figure 4 illustrates the comparative results obtained from the CPU and GPU. It demonstrates that when a GPU device was utilized, an increase in the number of processes up to 12 did not result in an overall duration increase.

7 Future Work

This work primarily covers the description of the model and the search for the optimal processing conditions. Given the existence of alternative strategies for optimizing the computational process, such as sorting the inputs by length first and subsequently dividing them into batches or limiting the number of threads to one and increasing the number of processes, further research on this topic is anticipated.

Another further objective is to evaluate the performance of the Langtok model in comparison with other language identification tools, such as langdetect [8], fasttext [5] [4], or langid [6].

8 Conclusion

The objective of this paper was to introduce the Langtok model and illustrate its potential applications. Additionally, a series of experiments was conducted with the aim to identify the optimal processing parameters. The experiments have demonstrated that the optimal batch size for the processing of multiple inputs in a simultaneous manner is within the range of four to ten. Finally, it has been found that when processing multiple input files simultaneously on a CPU device, it is extremely important to ensure that the number of computing resources is not exceeded. Otherwise, the efficiency of the processing will

Table 4: Statistics from test runs on CPU

number of processes	duration [s]	CPU memory / process	%CPU values / process
1	413	1,0g - 1,3g	1300 - 1620
2	242	1,0g - 1,1g	1560 - 1606
3	2764	970m - 1,1g	950 - 1160
4	2398	914m - 1,1g	750 - 840
6	2334	880m - 1,0g	507 - 572
12	1872	959m - 1,0g	256 - 277

Table 5: Statistics from test runs on GPU

number of processes	duration [s]	GPU memory [MB]	avg %GPU
1	181	1291,8	34,2
2	103	2232,5	58,4
3	72	2455,8	57,8
4	68	3896,4	79,7
6	54	4614,5	75,2
12	47	10053,9	92,0

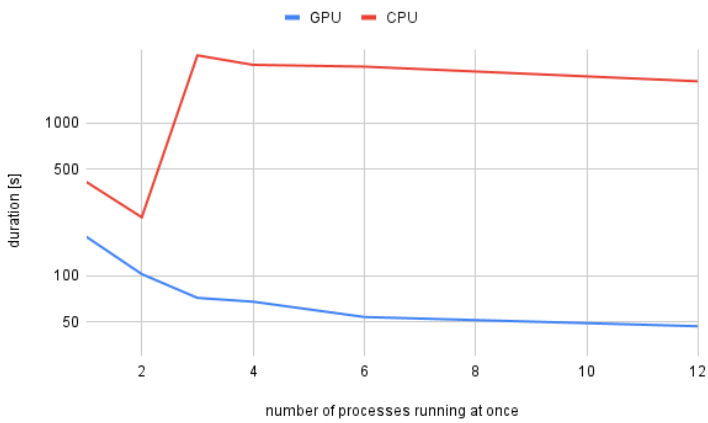


Fig. 4: Comparison of values measured on CPU and GPU

decrease significantly. Processing on a GPU device, however, did not pose such a threat, and the sole constraint that was identified was the limitation of memory resources.

Acknowledgements. The work described herein has been supported by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2023062 LINDAT/ CLARIAH-CZ

References

1. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
2. Costa-jussà, M.R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., et al.: No language left behind: Scaling human-centered machine translation. arXiv preprint arXiv:2207.04672 (2022)
3. Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
5. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
6. Lui, M., Baldwin, T.: langid. py: An off-the-shelf language identification tool. In: Proceedings of the ACL 2012 system demonstrations. pp. 25–30 (2012)
7. McCann, P.: fugashi, a tool for tokenizing Japanese in python. In: Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS). pp. 44–51. Association for Computational Linguistics, Online (Nov 2020), https://www.aclweb.org/anthology/2020.nlposs-1.7
8. Shuyo, N.: Language detection library for java (2010), http://code.google.com/p/language-detection/

WebMap: Improving LLM Web Agents with Semantic Search for Relevant Web Pages

Michal Spiegel[1,2] and Aleš Horák[1]

[1] Masaryk University
[2] Kempelen Institute of Intelligent Technologies

Abstract. The development of autonomous intelligent agents capable of complex decision-making and task-solving within specific environments remains a key challenge in Artificial Intelligence. This work focuses on autonomous agents navigating websites and solving tasks on the web through a browser, with experiments conducted in both English and Czech to evaluate performance across different linguistic contexts. It highlights the limitations of current state-of-the-art web agents, which lack in-domain knowledge about the websites they operate on and often fail to navigate to critical resources. To address this, we propose WebMap, a novel approach that preprocesses the website in advance and retrieves only task-relevant web pages that serve as a shortcut to essential resources. Experimental validation on the WebArena dataset demonstrates that WebMap achieves a 15% relative gain in task completion over the current leading method.

Keywords: autonomous web navigation, autonomous LLM agents, LLM reasoning, HTML understanding, vector databases.

1 Introduction

One of the key challenges in Artificial Intelligence has long been developing autonomous intelligent agents capable of complex decision-making and task-solving within specific environments. An agent is a system that operates independently, making decisions and taking actions to achieve specific goals within its environment without direct human intervention. These agents can perceive their surroundings, reason about information, and execute actions autonomously.

This work will focus on autonomous agents navigating websites and solving tasks on the web. In this case, the agent operates on the Internet through the use of a browser. The emphasis will be on web agents designed to function within a single website rather than general-purpose agents that work across the entire web.

An intelligent agent could effectively serve as a natural language interface to these web-based systems. That would mean the user could perform any task just by issuing a natural language command. This would streamline the

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 139–152, 2024.

process without requiring prerequisite knowledge or training from the user's side, making the interaction with the system much more effective and accessible.

A major flaw hindering the performance of the current state-of-the-art web agents is the lack of any in-domain knowledge about the website and its contents in advance. Because the agent lacks a comprehensive understanding of the entire website, it has a tendency to abandon tasks prematurely or search and explore the web pages using a trial-and-error strategy, which is ineffective and wastes time and resources. This happens because the model has no contextual information about the whole website and its functionalities and only ever sees isolated web pages, lacking a bigger picture view of the whole system.

This work proposes WebMap, a novel approach that improves web agents by providing more in-domain knowledge. WebMap preprocesses the website in advance, digesting and processing its contents into a semantic map of the whole website, all of its web pages, and functionalities. When confronted with a new task it retrieves only task-relevant URLs that serve as a shortcut for the agent to only the relevant web pages. This allows the agent to skip much of the work spent searching and exploring the website and effectively makes the task significantly easier.

We experimentally validate the proposed method on a subset of tasks from the WebArena dataset [30]. WebMap achieves 15% more tasks in relative gain over the state-of-the-art baseline method [15] and shows significant promise for further improvement.

The key contributions of this work are as follows:

- **Design and implementation of WebMap**, a novel method that improves the accuracy and efficiency of autonomous agents on the web
- **Experimental evaluation and analysis** of the proposed method on the WebArena benchmark
- **Error analysis** of WebMap and baseline approaches on the WebArena benchmark, highlighting important challenges for future work

2 Related Work

Recent advancements in autonomous web navigation have produced state-of-the-art techniques focused on empowering agents to execute tasks on real-world websites using natural language instructions. WebLINX [19] supports conversational web navigation, enabling agents to follow complex, dialogue-based instructions effectively. Synapse [29] leverages state abstraction and exemplar prompting to filter out task-irrelevant information and enhance multi-step decision-making. WebVoyager [11] uses multimodal inputs, including images and HTML, which enriches contextual understanding and improves interaction with dynamic web content. AutoWebGLM [15] optimizes page-level comprehension by simplifying HTML structures and utilizing reinforcement learning to refine the agent's navigation strategy.

Although each of these approaches innovates in areas like HTML simplification, multimodal input integration, and multi-step instruction-following, they all lack a persistent memory of a website's overarching structure. Consequently, these agents operate on isolated individual pages, relying on trial-and-error navigation rather than drawing on in-domain knowledge that could enable more streamlined, expert-level task completion.

By contrast, WebMap introduces a novel solution through the construction of a semantic map of the entire website, providing agents with comprehensive in-domain knowledge of its structure, contents, and functionalities. This preprocessed map allows WebMap to bypass trial-and-error navigation, enabling agents to access only task-relevant sections of a website. As a result, WebMap significantly enhances navigational efficiency and task success, marking a major step forward in autonomous web navigation.

3 WebMap - Converting Website HTML into a Vector Database

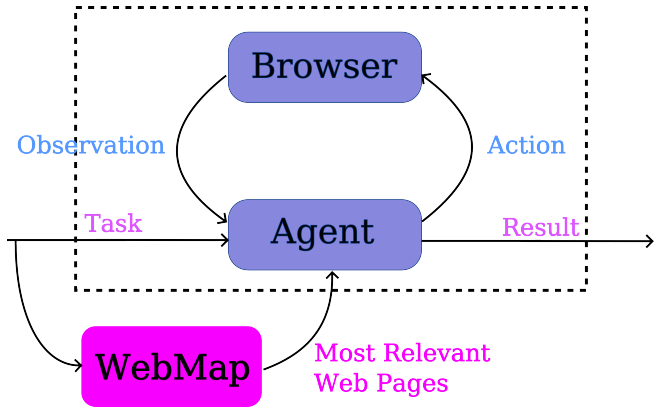


Fig. 1: The high-level workflow of the proposed solution. It takes the user task description as input and returns the most relevant web pages regarding the given task description.

Fig. 1 describes the high-level workflow of the proposed method. WebMap is completely separate and independent from the agent or the environment implementation. It works as a vector database storing representations of the individual web pages (URLs). For each new incoming task description for the agent, the task description is also used to search in the vector database for the most similar, most relevant web pages in regard to this task. Each web page is represented by its unique URL. The most relevant web pages are then passed to the agent which can use this information to start from a better position or

perform a more detailed analysis of the contents of the most relevant web pages to perform a more effective search through the possible web pages than would be possible without WebMap.

The four main components, which are outlined in the Fig. 2, are

- **Scraping** - this component is responsible for obtaining the raw, unprocessed web pages (HTML)
- **Preprocessing** - HTML web pages contain redundant elements, requiring cleaning and processing to achieve better representation.
- **Transforming data into a database** - to ensure effective search, it's necessary to convert them into representations that preserve the semantics and store these in a database that enables effective retrieval based on similarity or relevance

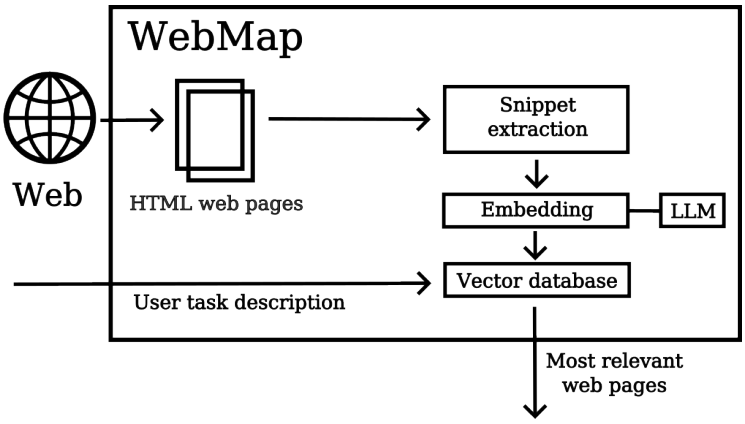


Fig. 2: WebMap and details its main components. WebMap first crawls the specified domain and retrieves a list of HTML web pages. Each HTML web page is then split into smaller chunks using snippet extraction and embedded using an LLM into a representation in vector space. The representations are used for indexing the snippets in a vector database. Given a user task description, this database is used to retrieve the most relevant web pages for completing the task.

The goal of WebMap is to convert raw data into a searchable database of web pages, allowing for efficient retrieval based on task descriptions. To achieve this, we represent text in vector space using state-of-the-art LLMs, which can encode semantic information. Preprocessing removes redundant elements like CSS styling, metadata, and unnecessary structural information to ensure efficient transformation and precise similarity search.

3.1 Preprocessing HTML Web Pages

Preprocessing HTML is crucial to reduce data volume and make similarity-based embedding with LLMs feasible, as typical web pages exceed model token limits. To streamline content, we:

- Remove non-semantic elements (e.g., metadata, stylesheets, scripts).
- Simplify structure by replacing tags (e.g., div, span) with inner content.
- Retain only visible attributes (e.g., type, placeholder).

Extracting Short and Meaningful Snippets We extract smaller snippets around salient elements, such as buttons and input fields, to simplify long HTML pages, adding surrounding context to clarify their function. [3]

3.2 Transforming Data into a Vector Database

After snippet extraction, we store them in a vector database to support similarity searches by converting snippets into vector representations for distance-based metrics like Euclidean and cosine similarity. The following sections detail embedding creation and vector database search.

HTML Snippet Embeddings Pre-trained LLMs generate high-dimensional vector embeddings for text, though they are not optimized for HTML, potentially making computation less efficient. While simpler methods like TF-IDF encode shallow semantics, specialized HTML models like DOM-LM [9], LayoutLM [28], MarkupLM [17] or HTLM [1] generally perform suboptimally against general LLMs [10]. HTML cleaning further simplifies text structure. We evaluate top models from the MTEB benchmark [23]:

- SFR-Embedding-Mistral [22]
- text-embedding-gecko [16]
- gte-large-en-v1.5 [18]
- LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-supervised [5]

To see a full evaluation and comparison of different embedding models and approaches, see Appendix C. Based on our evaluation, SFR-Embedding-Mistral with instruction turns out to be the most suitable model for this task.

4 Experimental Results & Discussion

This section presents the experimental results of the proposed WebMap method, assessing various implementations (e.g., different embedding techniques). Experiments were conducted in both English and Czech to test the method's effectiveness across linguistic contexts. However, due to the lack of existing Czech

[3] Inspired by [10]

datasets for web navigation evaluation, primary results focus on a widely used English dataset to maintain comparability with established research and benchmarks. Notably, this study also evaluated WebMap using Masaryk University's Information System (IS MU), a complex platform that supports a range of academic and administrative tasks and offers structured, semantically rich content in both English and Czech. Qualitative evaluations on the Czech IS MU interface, using multimodal versions of the pre-trained models, demonstrated correlations with human judgments comparable to those achieved on the English WebArena dataset.

4.1 WebArena Evaluation Environment and Dataset

Among recent datasets, WebArena uniquely offers live, self-hostable websites, enabling the mining of in-domain information. It includes about 800 complex, realistic tasks, highlighting the practical applicability of WebMap and addressing the challenges posed by real-world website complexity.

5 Baseline Agent

For a more credible evaluation and comparison, it is necessary to establish a baseline agent. We evaluate multiple different approaches. However, since only Gemini 1.5 Pro reaches non-zero accuracy, we do not provide any explicit evaluation results for these approaches. Notably, we have experimented with the following approaches as baseline agents:

- Heuristic rule-based agents (e.g. text similarity)
- LLM-based approaches
  - Using in-context learning
  - Different agent architectures
    - Recursively Criticizes and Improves (RCI) [13]
    - Chain-of-Thought (CoT) [27]
    - AutoWebGLM [15]
  - Different underlying LLMs
    - T5-based encoder-decoder models (FLAN-T5 [8], CoT fine-tuned variants [14])
    - Open-source decoder-only models (Vicuna [7], Mistral [12], Llama 2 [26], Llama 3 [2], web navigation fine-tuned variants [19])
    - Multimodal models (e.g. Fuyu [4], Gemini Pro Vision [3])
    - Gemini 1.5 Pro [3]

The only agent that was able to achieve moderate success on the WebArena dataset was a combination of AutoWebGLM, Gemini 1.5 Pro, few-shot learning and Chain-of-Thought prompting. The agent achieved an accuracy rate of 15% which is comparable to the best-performing WebArena evaluated GPT4-based agent which achieved 14% accuracy on the dataset. Therefore, the baseline agent is composed as follows:

- **AutoWebGLM** provides the agent with a simplified form of HTML as an observation space and defines the action space
- **Gemini 1.5 Pro** acts as the reasoning engine for input processing, planning and action prediction
- **In-context examples** are given to the model in the prompt to improve downstream performance
- **Chain-of-Thought** prompting [27] is used to improve reasoning abilities

5.1 End-to-end Evaluation of the Best-performing Strategy

To comprehensively evaluate the whole method, we perform end-to-end evaluation of WebMap on a subset of tasks from the WebArena dataset. The Table 1 shows results of multiple experiments done with different base models and compared to their version using WebMap. The results show that WebMap consistently improves the base models with a mean approximate relative gain of 30% over just using base models.

Table 1: The performance of WebMap on multiple base models.

Agent	Accuracy	Relative gain
AutowebGLM+Gemini	17%	–
AutowebGLM+Gemini+WebMap	20%	15%

6 Conclusion

Autonomous web navigation promises a revolution in human-computer interaction, providing more accessible user interfaces and streamlining complex processes with the use of intelligent agents. However, current autonomous web agents substantially fall behind in performance and effectiveness. Using in-context We analyse the current state-of-the-art (SOTA) methods of autonomous web navigation and find that in a significant number of cases, the agents completely fail to navigate to relevant web pages and resources important for completing the task on the web. This is confirmed by an error analysis of a SOTA agent on a subset of tasks from the WebArena dataset (Table 2).

This work explores a problem of effectively representing website contents to enable fast and accurate search for relevant resources. We propose WebMap, a solution for processing website contents into an information retrieval system that enables effective search for relevant web pages based on a user task description. The main problem lies in effective processing and understanding of HTML. We design the method, experimentally evaluate multiple alternatives and provide experimental validation on the WebArena evaluation dataset. WebMap achieves improved accuracy of over 15% in relative gain over state-of-the-art

baseline agent on real-world complex web navigation tasks from the WebArena dataset.

WebMap significantly impacts the current state of research on autonomous web navigation agents by proving the importance of modeling in-domain information as highlighted in the provided error analysis, including but not limited to representing website contents, standard interfaces, or usage of functionalities.

Acknowledgements. 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. Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G., Zettlemoyer, L.: HTLM: Hyper-Text Pre-Training and Prompting of Language Models (Jul 2021), http://arxiv.org/abs/2107.06955, arXiv:2107.06955 [cs]
2. AI@Meta: Llama 3 model card (2024)
3. et al., P.G.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context (2024), https://arxiv.org/abs/2403.05530
4. Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., Taşırlar, S.: Introducing our multimodal models (2023), https://www.adept.ai/blog/fuyu-8b
5. BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., Reddy, S.: LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders (Apr 2024), http://arxiv.org/abs/2404.05961, arXiv:2404.05961 [cs]
6. Chen, Q., Geng, X., Rosset, C., Buractaon, C., Lu, J., Shen, T., Zhou, K., Xiong, C., Gong, Y., Bennett, P., Craswell, N., Xie, X., Yang, F., Tower, B., Rao, N., Dong, A., Jiang, W., Liu, Z., Li, M., Liu, C., Li, Z., Majumder, R., Neville, J., Oakley, A., Risvik, K.M., Simhadri, H.V., Varma, M., Wang, Y., Yang, L., Yang, M., Zhang, C.: MS MARCO Web Search: a Large-scale Information-rich Web Dataset with Millions of Real Click Labels. In: Companion Proceedings of the ACM on Web Conference 2024. pp. 292–301 (May 2024). https://doi.org/10.1145/3589335.3648327, http://arxiv.org/abs/2405.07526, arXiv:2405.07526 [cs]
7. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), https://lmsys.org/blog/2023-03-30-vicuna/
8. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V., Wei, J.: Scaling Instruction-Finetuned Language Models (Dec 2022). https://doi.org/10.48550/arXiv.2210.11416, http://arxiv.org/abs/2210.11416, arXiv:2210.11416 [cs]
9. Deng, X., Shiralkar, P., Lockard, C., Huang, B., Sun, H.: DOM-LM: Learning Generalizable Representations for HTML Documents (Jan 2022), http://arxiv.org/abs/2201.10608, arXiv:2201.10608 [cs]

10. Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., Faust, A.: Understanding HTML with Large Language Models (May 2023). https://doi.org/10.48550/arXiv.2210.03945, http://arxiv.org/abs/2210.03945, arXiv:2210.03945 [cs]
11. He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., Yu, D.: WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models (Feb 2024), http://arxiv.org/abs/2401.13919, arXiv:2401.13919 [cs]
12. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7B (Oct 2023). https://doi.org/10.48550/arXiv.2310.06825, http://arxiv.org/abs/2310.06825, arXiv:2310.06825 [cs]
13. Kim, G., Baldi, P., McAleer, S.: Language Models can Solve Computer Tasks (Jun 2023), http://arxiv.org/abs/2303.17491, arXiv:2303.17491 [cs]
14. Kim, S., Joo, S.J., Kim, D., Jang, J., Ye, S., Shin, J., Seo, M.: The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. arXiv preprint arXiv:2305.14045 (2023)
15. Lai, H., Liu, X., Iong, I.L., Yao, S., Chen, Y., Shen, P., Yu, H., Zhang, H., Zhang, X., Dong, Y., Tang, J.: AutoWebGLM: Bootstrap And Reinforce A Large Language Model-based Web Navigating Agent (Apr 2024). https://doi.org/10.48550/arXiv.2404.03648, http://arxiv.org/abs/2404.03648, arXiv:2404.03648 [cs]
16. Lee, J., Dai, Z., Ren, X., Chen, B., Cer, D., Cole, J.R., Hui, K., Boratko, M., Kapadia, R., Ding, W., Luan, Y., Duddu, S.M.K., Abrego, G.H., Shi, W., Gupta, N., Kusupati, A., Jain, P., Jonnalagadda, S.R., Chang, M.W., Naim, I.: Gecko: Versatile Text Embeddings Distilled from Large Language Models (Mar 2024). https://doi.org/10.48550/arXiv.2403.20327, http://arxiv.org/abs/2403.20327, arXiv:2403.20327 [cs]
17. Li, J., Xu, Y., Cui, L., Wei, F.: MarkupLM: Pre-training of text and markup language for visually rich document understanding. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 6078–6087. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.420, https://aclanthology.org/2022.acl-long.420
18. Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards General Text Embeddings with Multi-stage Contrastive Learning (Aug 2023). https://doi.org/10.48550/arXiv.2308.03281, http://arxiv.org/abs/2308.03281, arXiv:2308.03281 [cs]
19. Lù, X.H., Kasner, Z., Reddy, S.: WebLINX: Real-World Website Navigation with Multi-Turn Dialogue (Feb 2024). https://doi.org/10.48550/arXiv.2402.05930, http://arxiv.org/abs/2402.05930, arXiv:2402.05930 [cs]
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
21. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, chap. Evaluation in information retrieval. Cambridge University Press (2008)
22. Meng, R., Liu, Y., Joty, S.R., Xiong, C., Zhou, Y., Yavuz, S.: Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog (2024), https://blog.salesforceairesearch.com/sfr-embedded-mistral/

23. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: MTEB: Massive Text Embedding Benchmark (Mar 2023). https://doi.org/10.48550/arXiv.2210.07316, http://arxiv.org/abs/2210.07316, arXiv:2210.07316 [cs]
24. 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. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
25. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models (Oct 2021), http://arxiv.org/abs/2104.08663, arXiv:2104.08663 [cs]
26. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open Foundation and Fine-Tuned Chat Models (Jul 2023). https://doi.org/10.48550/arXiv.2307.09288, http://arxiv.org/abs/2307.09288, arXiv:2307.09288 [cs]
27. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2024)
28. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1192–1200 (Aug 2020). https://doi.org/10.1145/3394486.3403172, http://arxiv.org/abs/1912.13318, arXiv:1912.13318 [cs]
29. Zheng, L., Wang, R., Wang, X., An, B.: Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control (Jan 2024). https://doi.org/10.48550/arXiv.2306.07863, http://arxiv.org/abs/2306.07863, arXiv:2306.07863 [cs]
30. Zhou, S., Xu, F.F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., Neubig, G.: WebArena: A Realistic Web Environment for Building Autonomous Agents (Oct 2023), http://arxiv.org/abs/2307.13854, arXiv:2307.13854 [cs]

A Error Analysis

Despite the encouraging results, experimental evaluation of WebMap still shows a considerable error margin. For this reason, we decide to conduct an error analysis to understand the underlying issues and identify the main problems. The error analysis of both the baseline method and WebMap method with human annotations from the experiments in Section B can be seen in Table 2 and 3. This error analysis has been manually evaluated on the results of

the evaluation of the baseline agent and WebMap with human annotations discussed in Section B. The agents were evaluated on a sample of 63 tasks from the WebArena dataset. The occurrences do not sum up to 100% because many of these errors happen simultaneously in the same tasks, and it is not possible to evaluate certain tasks with only one type of error. The log files from both the evaluation and the error analysis are provided for inspection in attachments.

We identified the following major categories of errors:

- **Controls Understanding** captures all the situations in which the agent successfully navigates to the correct page, understands the task and what is needed to do, but ultimately fails to operate the needed controls correctly. For example, agents often struggle with formatting dates correctly and often produce the incorrect format or hallucinate invalid dates. Agents often forget to perform the last step to submit, show, or confirm selected settings.
- **Incorrect Web Page** is the second most predominant issue. This type of errors is what WebMap tries to solve. Agents have a tendency to navigate incorrectly and do not even find the correct resource.
- **Observation Understanding** describes a situation when the agents needs to extract some information from the observation (e.g. from the HTML), the information is perfectly visible, but the agents fails to recognize it
- **Planning and Reasoning Errors** often occur together often on tasks that require long-horizon planning, the agent forgets a vital step, makes an obvious reasoning error
- **Hallucinations** happen when the agent just straightforward outputs nonsense that just looks good, e.g., says it has successfully finished when it has not done any action at all
- **Evaluation Dataset Fault and not enough information in the observation** are both errors that are not directly the fault of the agent, e.g. there is a mistake in the dataset, in the observation
- **Fail to recover from an error** happens when the agent tries a strategy, the strategy fails and it just gives up, the inability to step back, rethink the problem and try again.

Table 2: The error analysis of the baseline AutowebGLM using Gemini 1.5 Pro.

Type of Error	Absolute Count	Percentage (%)
Controls Understanding	24	30
Incorrect Web Page	18	23
Observation Understanding	16	20
Reasoning Error	10	13
Planning Error	5	6
Hallucinations	4	5
Evaluation Dataset Fault	1	1
Not enough information in the observation	1	1
Fail to recover from an error	1	1

Table 3: The error analysis of AutowebGLM using Gemini 1.5 Pro together with human-annotated WebMap (Section B)

Type of Error	Absolute Count	Percentage (%)
Controls Understanding	31	40
Incorrect Web Page	0	0
Observation Understanding	10	13
Reasoning Error	6	8
Planning Error	7	9
Hallucinations	2	3
Evaluation Dataset Fault	3	4
Not enough information in the observation	1	1
Fail to recover from an error	7	9

The error analysis highlights the issue of control understanding that causes problems in 30-46% of tasks. Agents often have problems with operating more complex system interfaces that require multiple steps before the result is shown. They often struggle with correctly formatting dates. Similarly, they have problems with understanding simple protocols for working with certain interfaces, e.g., they often forget last steps such as submitting or confirming changes. At the same time, they often lack a good understanding of the observation. They often incorrectly reason about the observation and miss key information in the observation. We observe that in non-trivial amount cases, the HTML simply does not provide enough information to make the correct decision, even for a human.

B Evaluation using human annotations

Evaluating WebMap solely on its end-to-end performance in web navigation tasks (e.g., WebArena datasets) limits the ability to analyze its individual components. Thus, it is essential to separately evaluate the information retrieval task.

While there are various datasets for benchmarking information retrieval techniques [25], including those for question answering and web search [6], searching web pages based on task descriptions presents unique challenges. Although web search is similar, differences in query formats and document types complicate direct comparisons. Therefore, we manually annotated a subset of tasks from the WebArena dataset with relevant URLs that align with WebMap's output. This human-annotated gold standard serves two main purposes:

- **Determine Optimal Performance Increase:** Evaluate whether the benefits of improving this method outweigh the challenges.
- **Benchmark Automatic HTML Processing Techniques:** Comparing different HTML preprocessing strategies requires a gold standard for effective evaluation.

We manually annotate 184 tasks from the WebArena evaluation dataset. Each annotation represents the URL of the most relevant web page, the best starting point for successfully solving the current task.

B.1 Determining the Best Possible Performance Increase

Implementing WebMap presents challenges in HTML understanding and web page information retrieval. Before allocating resources, it's essential to assess its potential for enhancing end-to-end performance in web navigation tasks. To this end, we designed an experiment comparing a baseline AutowebGLM agent with a WebMap agent, using human annotations instead of automated HTML processing.

The baseline agent, based on AutoWebGLM, incorporates a few in-context examples and chain-of-thought prompting, utilizing the Gemini 1.5 Pro LLM. The WebMap agent differs only in that it begins tasks at the URLs specified in the human annotations and is prompted that the current web page is most relevant. This experiment evaluates a subset of 63 tasks from the WebArena dataset, chosen due to computational constraints. The results will gauge WebMap's feasibility in improving task completion.

Table 4: Performance of WebMap using human annotations instead of automatic processing strategies.

Agent	Accuracy	Relative Gain
AutowebGLM+Gemini	17%	–
AutowebGLM+Gemini+WebMap (using human annotations)	23%	$\sim$ 30%

Table 4 shows that the agent guided by human annotations completed 6

B.2 Automatic HTML Content Processing Strategies

Section B.4 details design decisions regarding specific WebMap components that require experimental evaluation, including:

- **Embedding Function:** This computes HTML snippet representations. Research indicates conventional LLMs (e.g., T5 [24]) outperform specialized models (e.g., MarkupLM [17]) [10]. The best-performing embedding models are selected based on the Massive Text Embedding Benchmark (MTEB) [23] and evaluated against the human-annotated gold standard.
- **Hyperparameters for Snippet Extraction:** A hyperparameter search is necessary to identify optimal settings, such as the maximum length and tree height of HTML snippets. The impact of snippet length and nested elements on retrieval performance remains unclear, as LLMs were not primarily trained on HTML modeling. Experimental evaluation against the gold standard is the best approach to address this.

B.3 Evaluation Methodology

To evaluate automatic processing strategies, we use the Mean Reciprocal Rank (MRR) metric, referencing a human-annotated gold standard. This choice is based on WebMap's role as an information retrieval system, which retrieves the most relevant web pages based on user task descriptions [20, p. 1]. Effectiveness is traditionally measured against a gold standard that defines the relevance of documents for specific queries [21]. In this case, each task description corresponds to a single relevant web page, simplifying the relevance assessment.

Selecting appropriate metrics is crucial, as traditional measures like precision and recall are less informative when only one relevant document exists. Since WebMap outputs a ranked list of web pages, the effectiveness is best captured by evaluating the rank of the human-annotated gold standard in this list. MRR is a suitable choice, as it assesses the mean rank of the first relevant document.

B.4 Effective Representation of HTML Snippets Using LLMs

This section examines methods for representing HTML snippets in vector space, crucial for WebMap's retrieval performance. While specialized architectures often struggle with HTML [10], conventional LLMs (e.g., T5 [24]) show promise in handling complex structures. We evaluate a subset of embedding techniques using the MRR methodology, with results shown in Table 4.

Based on evaluations from the Massive Text Embedding Benchmark (MTEB) [23], we selected techniques from relevant categories like *Overall* and *Retrieval*. The preliminary results suggest that SFR-Embedding-Mistral and gte-large-en-v1.5 show the most potential, with MRRs indicating the gold standard typically ranks within the top 3. However, high standard deviation points to performance instability, prompting further optimization of snippet extraction hyperparameters.

B.5 Embedding with Instruction

Some embedding models, like LLM2Vec [5], suggest adding instructions to the input text to enhance performance. Testing this with the top-performing SFR-Embedding-Mistral, we found that including an instruction significantly improved the MRR to 0.63. However, this also increased computation time due to the added prompt length.

Improving Layout Analysis of Scanned Invoices Using Line Detection

Hien Thi Ha[1,2]

[1] Natural Language Processing Center, Masaryk University, Brno, Czech Republic
[2] Le Quy Don Technical University, Hanoi, Vietnam
hienht@lqdtu.edu.vn

Abstract. The combination of text and visual information, such as layout and image features, has recently become dominant in the processing of visually rich documents. However, they are limited to word features, i.e., word position, font, and style. In this work, we propose the use of straight lines (or separators) to improve layout analysis and document understanding of invoices. Our preliminary evaluation with a dataset of Czech invoices shows the improvement in layout construction and, consequently, in information extraction in certain cases.

Keywords: invoice processing, layout analysis, line detection.

1 Introduction

The analysis of document structure can be divided into the *layout structure* (also called physical structure or geometric layout structure) and the *logical structure*. Geometric layout analysis of a document image generally involves different steps [12]:

- Binarization
- Noise removal
- Skew correction
- Page segmentation
- Zone classification
- Reading order determination

in which the exact order of steps varies between algorithms, and some algorithms might skip one or more of these processes, or apply them in a hybrid way. The first three stages are often considered pre-processing steps in layout analysis. The last step is generally considered a post-processing step. The remainder, including page segmentation and zone classification, are core parts of geometric layout analysis. The purpose of page segmentation is to divide the document image into homogeneous zones, each consisting of only one physical layout structure (text, graphics, images,...), and the latter, zone classification, classifies each zone into respective categories.

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2024, pp. 153–164, 2024.

In form-based documents such as invoices and receipts, information is often organized into groups. For example, an invoice might have a title, general information (including invoice number, invoice date, purchase order number, and order date), seller/buyer information (including company name, address, VAT number, and contact information), delivery address (including company name or contact person, address), table of items, payment information (including payment method, bank account, due date). These blocks are often separated by either spaces or straight lines. Recognizing these blocks of text is therefore important for invoice understanding.

Our previous works [4,5] show that layout analysis and text block recognition, in particular, significantly affects logical analysis and information extraction. While Tesseract OCR [13] gives us good character recognition and line segmentation results, it ignores visual features such as straight lines that separate blocks of information, for example, party blocks with other logical blocks and between party blocks. These separators were removed in pre-processing steps of Tesseract OCR.

Recently, document understanding has shifted from text-level manipulation to combining text and visual features. One of the most successful models is LayoutLM [15] (and its improved versions LayoutLMv2 [14] and LayoutLMv3 [7]). The LayoutLM model, inspired by BERT [2], jointly learned both text embeddings andal position embeddings (word's bounding box), and then added an image embedding layer (see Figure 1 for the architecture of the LayoutLM model). The document image is split into pieces, each of which has a one-to-one correspondence with the words based on the bounding box of each word from OCR results. The token image embeddings are generated from chunks using the Faster R-CNN [10] model. The entire document image is also fed into the Faster R-CNN model to produce [CLS] token for document image classification. For other tasks such as form understanding, only word's visual features are used.

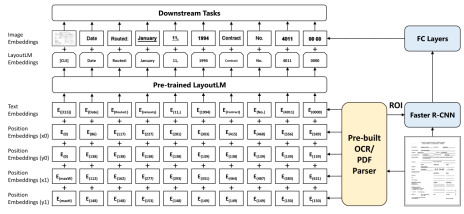


Fig. 1: LayoutLM architecture [15]

In [9], the authors located enclosing boxes by detecting line intersections and lines. The boxes were then combined with text clusters found by connected-component analysis to generate possible locations of keywords and data in the invoice before doing any OCR based on criteria, such as only those clusters containing a single or few words that are separated from other clusters by

a significant amount of white space are considered as a possible keyword. However, not all invoices contain such regular box structures and both keyword and data may be contained in the same enclosing box.

OCRMiner [5] is a system designed to automated processing of business documents such as invoices and contracts based solely on the analysis of the OCR processing of the document image. The document is represented as a graph of hierarchical text blocks built up from words or lines resulting from the OCR engine in a bottom-up approach. In this paper, we discuss how visual components, particularly straight lines can help in building the physical layout of the invoice and consequently help improve the result of further modules in the OCRMiner pipeline.

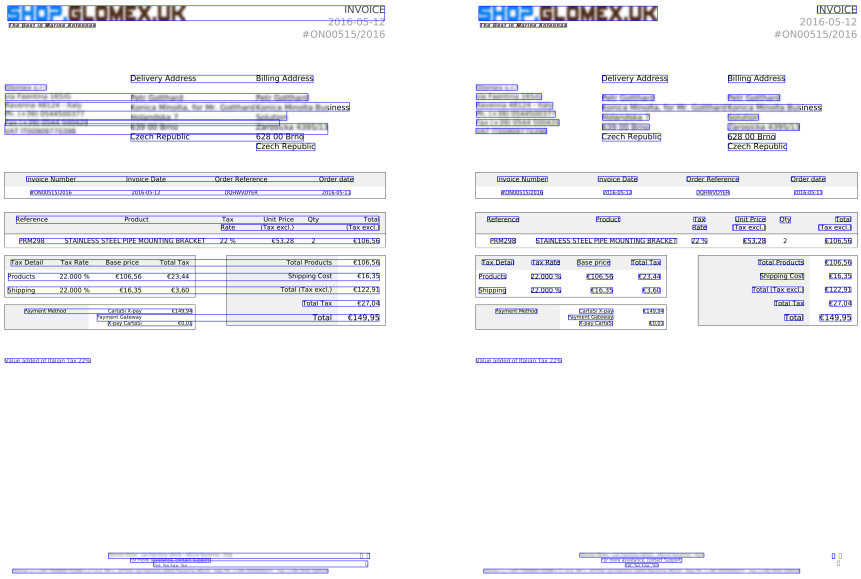
2 Layout analysis in scanned invoices

Unlike other documents, such as books, magazine publications, and newspapers, the layout analysis of invoices faces several challenges. Firstly, the layout of invoices is complex and varies greatly from vendor to vendor. The second is the ambiguous block boundary, where the boundary between blocks is not as clear as in other document types, such as scientific papers. Last but not least, ambiguous text blocks and tables due to the fact that information in invoices is often presented in tabular form.

In the OCRMiner system, we use Tesseract OCR to capture the text and initial layout information, such as the word's bounding box and the line's bounding box. We first look at the output from Tesseract OCR on scanned invoices using different page segmentation algorithms. Tesseract currently supports 14 page segmentation modes [3]:

- 0 Orientation and script detection (OSD) only.
- 1 Automatic page segmentation with OSD.
- 2 Automatic page segmentation, but no OSD or OCR. (not implemented)
- 3 Fully automatic page segmentation, but no OSD. (Default)
- 4 Assume a single column of text of variable sizes.
- 5 Assume a single uniform block of vertically aligned text.
- 6 Assume a single uniform block of text.
- 7 Treat the image as a single text line.
- 8 Treat the image as a single word.
- 9 Treat the image as a single word in a circle.
- 10 Treat the image as a single character.
- 11 Sparse text. Find as much text as possible in no particular order.
- 12 Sparse text with OSD.
- 13 Raw line. Treat the image as a single text line, bypassing Tesseract-specific hacks.

[3] "https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html#page-segmentation-method"



(a) using page segmentation 3 (b) using page segmentation 11

Fig. 2: Example of text lines detected by Tesseract OCR using different page segmentations

In these modes, mode 0 is to detect the orientation of the document, and mode 3 is the default which builds a full hierarchical layout of the document, i.e., treating the text as a proper page with multiple words, lines, and paragraphs, but no orientation/script detection is performed. To get this information, we need to run Tesseract twice, first run with –psm 0 to detect the OSD, then again with –psm 3 to OCR the actual text. In this mode, Tesseract tends to merge lines horizontally, regardless of how large the space between columns is, as long as the text style is similar (see Figure 2). This can be accurate for general information (e.g., invoice number, dates, payment method,...). However, merging party blocks (i.e., buyer and seller) violates the reading order.

Mode 4 is useful when we need to OCR column data and require text to be concatenated row-wise, such as receipts or tables. Mode 5 is similar to mode 4 but for rotated images. Mode 6 is for uniform text, i.e., text in a single font face, which is best for simple books. When we work with a single line of uniform text, e.g a vehicle registration plate, page segmentation mode 7 should be utilized. Mode 8 can be used to detect a single word of uniform text in an image, e.g., the name of a store on the storefront photo. Mode 9 seems to be for text wrapped around an (invisible) arc region, while mode 10 is used when we have already extracted every single character from the image.

Mode 11 and 12 are for sparse text. These are suitable when a piece of text is closely cropped, the text is stylized, or Tesseract-OCR may not automatically recognize its font. In these modes, Tesseract creates only lines of text, although it also has blocks, each containing only 1 line. This segments lines well, although in some cases, it still merges columns when the text is too close (see Figure 2, the second line in the delivery address is merged with the second line in the billing address), and is inconsistent in some cases. In the following example 3, on the left column, the issued date keyword and data are merged, whereas the due date and the payment date are not.

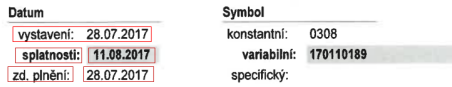


Fig. 3: Example of inconsistency in Tesseract's page segmentation where keyword and data are merged in the first red box, but in the two boxes below, they are not.

Because invoices are sparse text, in OCRMiner, we use page segmentation 11 to create lines, then group lines into blocks if they are aligned, have similar styles, and their distance is less than a predefined threshold.

In form-like documents, blocks of information are often separated using either spaces or horizontal and/or vertical lines. However, Tesseract OCR and previous works in this field ignored these pivots. In the next sections, we will present how to detect and use these lines to build the physical layout.

3 Line detection and applying detected lines for invoice layout analysis

3.1 Line detection

Hough Transform (HT) was introduced by Paul Hough [6] in a patent filed in 1962. It is an ingenious method that transforms a global curve detection problem into an efficient peak detection problem in parameter space. HT was brought to the attention of the mainstream image-processing community by Rosenfeld [11]. To detect straight lines, Hough chose the slope-intercept parametric equation of the line:

$$f(x, y) = y - mx - c = 0 \tag{1}$$

where the parameters m and c are slope and y-intercept of the line respectively. Unfortunately, this method is sensitive to the choice of coordinate axes on the image plane because both the slope and the intercept are unbounded. To resolve this problem, Duda and Hart [3] suggested to parameterize straight

line by the length, ρ, and orientation, θ, of the normal vector to the line from the image origin (illustrated in Figure 4):

$$xcos\theta + ysin\theta = \rho \tag{2}$$

with θ is in the interval $[0, \pi)$.

Notice that each point (x_i, y_i) in the image can be transformed into a sinusoidal curve in the θ, ρ plane defined by:

$$\rho = x_i cos\theta + y_i sin\theta \tag{3}$$

The curves corresponding to collinear points in the image intersect at a common point, let's say (θ_0, ρ_0). This transforms collinear points in the image space into concurrent curves in the Hough space. The Hough transform constructs a histogram array representing the parameter space (i.e., an $M \times N$ matrix, for M different values of the radius ρ and N different values of θ. The size of the matrix depends on the accuracy we need). For each parameter combination, ρ and θ, we then find the number of non-zero pixels in the input image that would fall close to the corresponding line and increment or vote up the array at position (ρ, θ) appropriately. The local maxima in the resulting histogram indicate the parameters of the most probable lines. Kiryati at al. [8] suggested Probabilistic Hough Transform to replace the full-scale voting in the incrementation stage with a limited pool of m ($<$ M) randomly selected object pixels to reduce the algorithm's complexity.

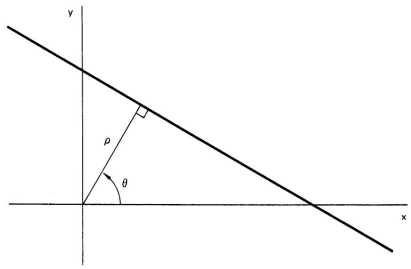


Fig. 4: The normal parameter for a line [3]

3.2 Applying detected lines to invoice layout analysis

To detect lines in invoice images using Probabilistic Hough Transform (probabilistic_hough_line) function in OpenCV, we first preprocess the image by converting it into gray, then performing Canny edge detection [1]. In our experiment, we filter out lines with length less than a threshold (set to 100 by experiment) to avoid noise in the image. Figure 5 illustrates lines detected from an invoice image using Probabilistic HT and Figure 6 shows various invoice layouts

just by detected lines. These layouts can be used in document image classification, or in classifying the invoice into known and unknown categories for later processing.

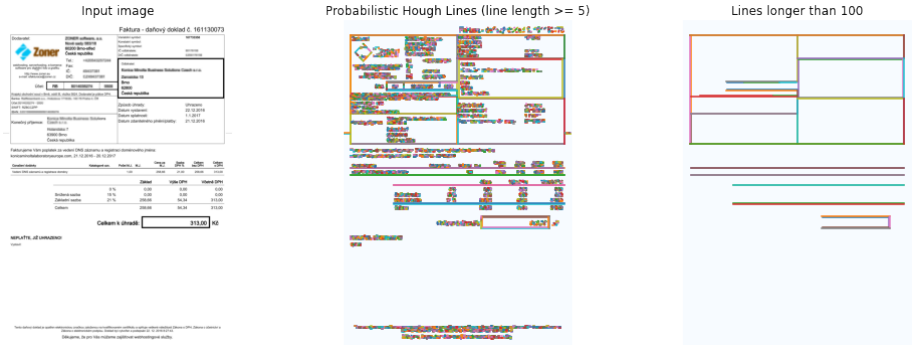


Fig. 5: Example of lines detected in an invoice image using Probabilistic Hough Transform with different length thresholds

From the above layout examples, we observe that:

- Information in enclosed boxes often belongs to the same class (e.g., seller, buyer). In some cases, a horizontal merge inside the box is acceptable (e.g., keyword and data), but in some other cases, a horizontal merge violates reading order (e.g., "Konečný příjemce:" and "Czech s.r.o." are horizontal alignment in the invoice in Figure 5 but should not be merged.)
- Information between two long horizontal lines often belongs to the same big categories (e.g., general info, bank info, parties) but may not belong to the same classes (e.g., seller and buyer), so can merge vertically but not horizontally.
- Seller information can be fragmented vertically or horizontally at the top of the page and separated from other blocks by a horizontal line.
- Short or middle long horizontal lines can be used to separate keywords and data

Based on that observation, to build text blocks, we use the same merging lines into blocks tactic described in [5] but add a condition, only merge if there is no detected line between them. Figure 8 and Figure7 show an example of a layout with and without recognized lines. The blocks detected by both modules are shown in brown, and the differences between the two modules are marked in blue and red boxes, respectively.

Without using lines, the bank account information group was merged with two groups below, one of which was the delivery address. The merge also happened with groups containing variable symbol and date, and between price and subtotal groups.

Fig. 6: Layout examples using only detected lines

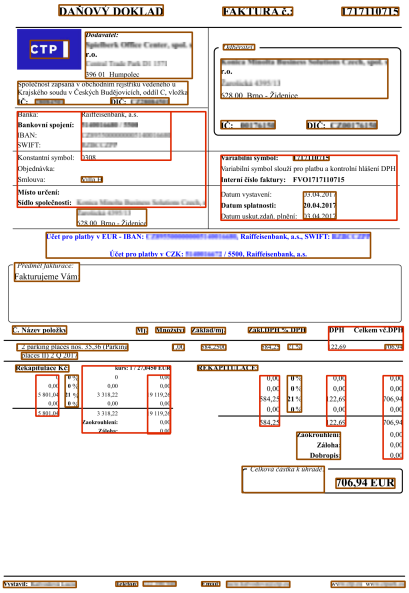


Fig. 7: Text blocks without using detected lines.

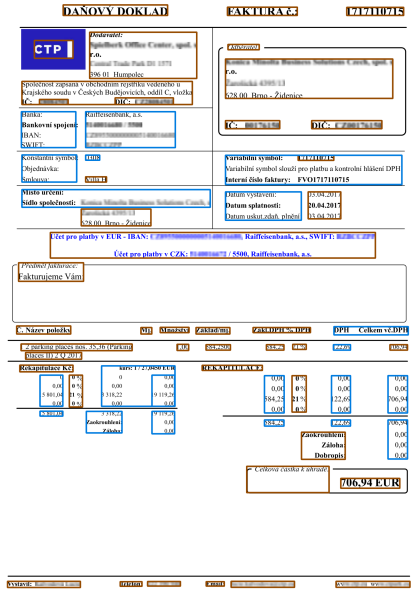
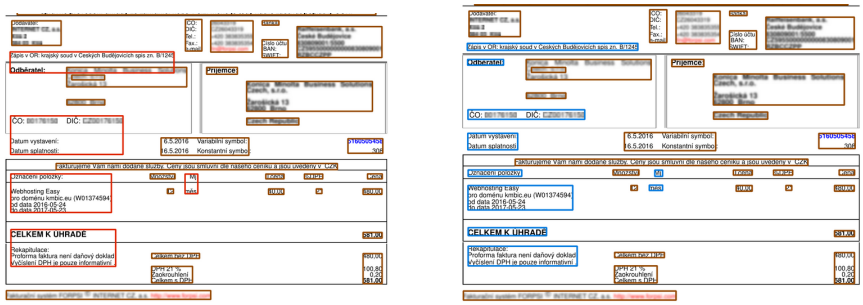


Fig. 8: Text blocks using detected lines.

The preliminary evaluation of the information extraction task shows that layout using detected lines improves the result in certain cases. First, detected lines prevent grouping lines in different categories into the same block. For example, in Figure 10, the seller information was merged with buyer information, plus Tesseract-OCR misrecognized the keyword *Kupující*/*Purchaser*, resulting in the whole block being assigned as *seller* only and buyer information being missed. In another case (see Figure 9), the keyword *Odběratel*/*Customer* is grouped with the above line. Then, the alignment between the keyword for the buyer information and its data is lost, affecting the extraction process. A similar problem happens with the customer's company ID and VAT number when they are merged with the invoice date and due date block of that invoice, or in the invoice in Figure 11.

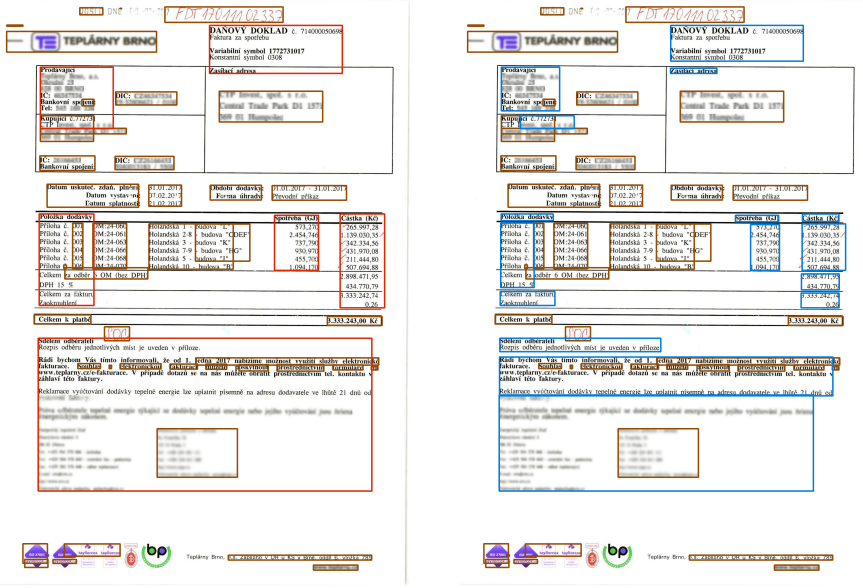
4 Conclusion

In this paper, we propose a simple yet potential visual feature to be used for layout analysis of form-like document images such as invoices. The straight lines detected by the Probabilistic Hough Transform form the invoice layout, which can be useful for document image classification problems. Our preliminary evaluation shows that using these lines as a barrier in constructing the invoice layout improves the information extraction result in certain cases. The full



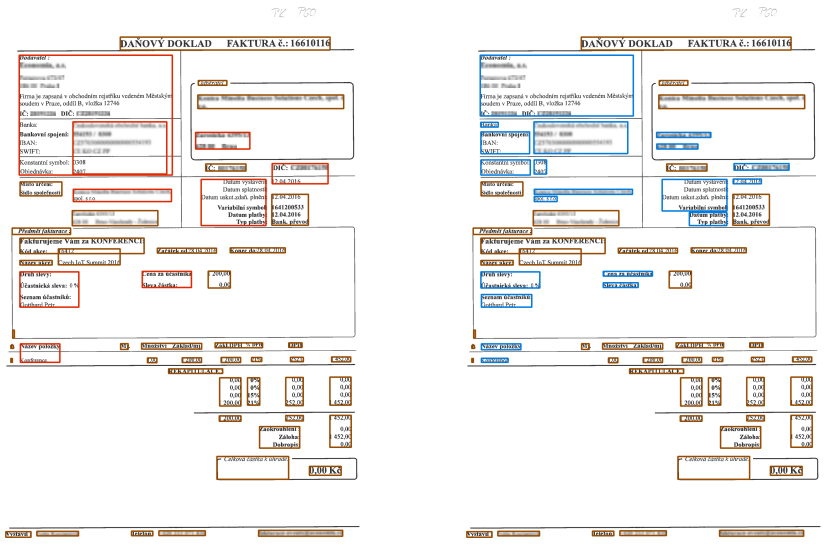
(a) Text blocks without straight lines (b) Text blocks using straight lines

Fig. 9: Example of improvement in information extraction (1).



(a) Text blocks without straight lines (b) Text blocks using straight lines

Fig. 10: Example of improvement in information extraction (2).



(a) Text blocks without straight lines (b) Text blocks using straight lines

Fig. 11: Example of improvement in information extraction (3).

evaluation will be done in future work. This work only uses straight lines in a small step of the physical layout analysis process. We would like to further explore the potential of this feature. For example, grouping the text in a closing box that does not violate the reading order.

References

1. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM **15**(1), 11–15 (1972)
4. Ha, H.T., Medved', M., Nevěřilová, Z., Horák, A.: Recognition of ocr invoice metadata block types. In: Text, Speech, and Dialogue: 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018, Proceedings 21. pp. 304–312. Springer (2018)
5. Ha, H.T., Horák, A.: Information extraction from scanned invoice images using text analysis and layout features. Signal Processing: Image Communication **102**, 116601 (2022)
6. Hough, P.V.: Method and means for recognizing complex patterns (Dec 18 1962), uS Patent 3,069,654

7. 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)
8. Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic hough transform. Pattern recognition **24**(4), 303–316 (1991)
9. Kosiba, D.A., Kasturi, R.: Automatic invoice interpretation: invoice structure analysis. In: Proceedings of 13th International Conference on Pattern Recognition. vol. 3, pp. 721–725. IEEE (1996)
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1137–1149 (2016)
11. Rosenfeld, A.: Picture processing by computer. ACM Computing Surveys (CSUR) **1**(3), 147–176 (1969)
12. Shafait, F.: Geometric Layout Analysis of scanned documents. Ph.D. thesis, Technische Universität Kaiserslautern (2008)
13. Smith, R.W.: Hybrid page layout analysis via tab-stop detection. In: Document Analysis and Recognition, 10th International Conference on. pp. 241–245. IEEE (2009)
14. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., et al.: Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. arXiv preprint arXiv:2012.14740 (2020)
15. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1192–1200 (2020)

Subject Index

Author Index

RASLAN 2024

Eighteenth 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 2024

ISBN 978-80-263-1835-4
ISSN 2336-4289