CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF CYBERNETICS
CZECH INSTITUTE OF INFORMATICS, ROBOTICS AND CYBERNETICS

From FastText to Transformer Models, and their Application in Retrieval-Augmented Generation

Bachelor's Thesis

Yauheni Zviazdou

Prague, May 2024

Study programme: Open Informatics
Branch of study: Artificial Intelligence and Computer Science

Supervisor: Ing. Jan Šedivý, CSc.

Acknowledgments

Firstly, I would like to express my gratitude to my supervisor.

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Zviazdou Yauheni** Personal ID number: **507333**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

From FastText to Transformer Models, and their Application in Retrieval-Augmented Generation

Bachelor's thesis title in Czech:

Od FastText k Transformer model m a jejich aplikace v Retrieval-Augmented generování

Guidelines:

Review the representation of words, sentences, and paragraphs, progressing from traditional methods like FastText to advanced transformer-based models such as BERT. The primary focus is to evaluate selected representations using analogy tests and confusion matrix. Use the UPV corpus set for evaluation.
In the second part of the study, emphasis will shift towards selecting optimal representations for Retrieval-Augmented Generation (RAG) algorithms. The investigation will determine the most efficient embeddings and optimal text chunk size for question-answering tasks, particularly in the context of natural language answers generation from technical manuals. Conduct a comprehensive evaluation with a particular focus on suggesting an optimal representation model that balances factuality and CPU requirements.

Bibliography / sources:

[1] FastText documentation and tutorials, https://fasttext.cc/
[2] LangChain documentation, https://js.langchain.com/docs
[3] Word2Vec tutorials, https://www.tensorflow.org/text/tutorials/word2vec

Name and workplace of bachelor's thesis supervisor:

Ing. Jan Šedivý, CSc.   Big Data and Cloud Computing CIIRC

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **02.01.2024**   Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

_____ _____ _____
Ing. Jan Šedivý, CSc. prof. Dr. Ing. Jan Kybic prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature Head of department's signature Dean's signature

III. Assignment receipt

_____ _____
Date of assignment receipt Student's signature

Declaration

I declare that presented work was developed independently, and that I have listed all sources of information used within, in accordance with the Methodical instructions for observing ethical principles in preparation of university theses.

Date ....

Abstract

This thesis investigates the application of word and sentence embeddings in Retrieval-Augmented Generation (RAG) for factual Question Answering (QA) tasks using technical manuals. The study explores the effectiveness of traditional FastText embeddings and advanced transformer-based models like Bidirectional Encoder Representations from Transformers (BERT) in capturing semantic relationships within text. We evaluate the quality of these representations using analogy tests and confusion matrix analysis on the UPV corpus set.

Subsequently, we will select optimal representations for RAG algorithms and assess their impact on factual accuracy and computational efficiency during QA. By analyzing the performance with different text chunk sizes, we aim to identify the optimal configuration for factual RAG in technical domains. This research contributes to the field of Natural Language Procession (NLP) by providing insights into selecting effective representations that balance factual accuracy and computational efficiency for QA systems.

Keywords NLP, Word Embedding, Transformers, RAG, QA, Semantic Textual Similarity (STS)

Abstrakt

Tato práce zkoumá aplikaci embeddingů slov a vět v modelu Retrieval-Augmented Generation (RAG) pro úlohy Question Answering (QA) zaměřené na fakta s využitím technických příruček. Studie se zabývá účinností tradičních embeddingů FastText a pokročilých modelů založených na transformátoru, jako je Bidirectional Encoder Representations from Transformers (BERT), při zachycování sémantických vztahů v textu. Kvalitu těchto reprezentací hodnotíme pomocí analogových testů a analýzy confusion matrix na korpusu UPV. Následně vybereme optimální reprezentace pro algoritmy RAG a posoudíme jejich vliv na faktickou přesnost a výpočetní efektivitu během QA. Analýzou výkonu s různými velikostmi textových fragmentů se snažíme identifikovat optimální konfiguraci pro faktické RAG v technických oborech. Výzkum přispívá k oblasti zpracování přirozeného jazyka (Natural Language Procession (NLP)) tím, že poskytuje poznatky o výběru efektivních reprezentací, které vyvažují faktickou přesnost a výpočetní efektivitu pro systémy QA.

Klíčová slova NLP, Word Embedding, Transformátory, RAG, QA, Sémantická podobnost textu

Abbreviations

RAG Retrieval-Augmented Generation

NLP Natural Language Procession

STS Semantic Textual Similarity

QA Question Answering

BERT Bidirectional Encoder Representations from Transformers

CBOW Continuous Bag-of-Words

GloVe Global vectors

ML Machine Learning

NN Neural Network

mBERT Multilingual Bidirectional Encoder Representations from Transformers

MTEB Massive Text Embedding Benchmark

ALBERT A Lite BERT

SimCSE Simple Contrastive Learning of Sentence Embeddings

RetroMAE Retrieval-oriented Language Models Via Masked Auto-Encoder

mE5 Multilingual E5

Contents

1 Introduction

1.1 Text representation

The human language, with its nuances and complexities, presents a significant challenge for machines to understand. Natural Language Procession (NLP) bridges this gap, and at its core lies the critical concept of text representation. This process acts as a translator, bridging the gap between the richness of text and the numerical language that machines understand. By effectively capturing the meaning within words and their relationships, text representation empowers NLP models to leverage machine learning's capabilities. From sentiment analysis to machine translation, this ability to represent meaning fuels the advancements in NLP, enabling machines to interact with and decipher human language with ever-increasing accuracy.

1.2 Evolution of text representation methods

NLP has undergone a significant transformation in its approach to text representation. Early methods, such as one-hot encoding, while simple to implement, suffered from limitations in efficiency due to dimensionality and sparsity issues.

Word embedding techniques (e.g., Word2Vec, GloVe, FastText) offered a significant improvement by capturing semantic relationships between words through high-dimensional word vectors. However, these techniques primarily focused on local context within a limited window, hindering their ability to capture complex relationships within sentences or documents.

The emergence of deep learning architectures, particularly transformer-based models like Bidirectional Encoder Representations from Transformers (BERT), revolutionized the field of text representation. These models allows to not only understand the meaning of individual words but also consider their interaction and context within a sentence or document.

1.3 Research objective

This research aims to evaluate the effectiveness of various word, sentence, and paragraph representations for their subsequent application in Retrieval-Augmented Generation (RAG) algorithms, with a specific focus on the domain of technical Question Answering (QA).
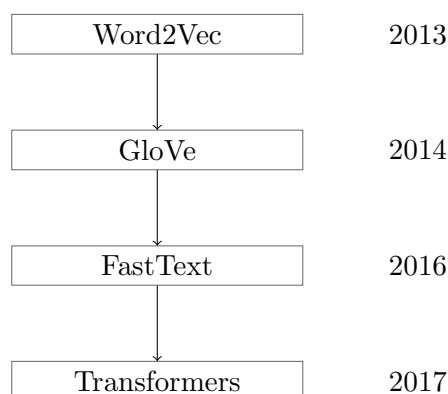
Word2Vec	2013
GloVe	2014
FastText	2016
Transformers	2017

Figure 1.1: Evolution of the text representation methods.

2 Literature Review

2.1 Traditional word embedding methods

Word2Vec [15]

Word2Vec is algorithm that generates word embedding using information about target word (context). Word2Vec uses Neural Network (NN) and Machine Learning (ML) techniques to generate word embedding for every word in vocabulary during training. As NN architecture are used Continuous Bag-of-Words (CBOW) and Skip-gram, Fig. 2.1.
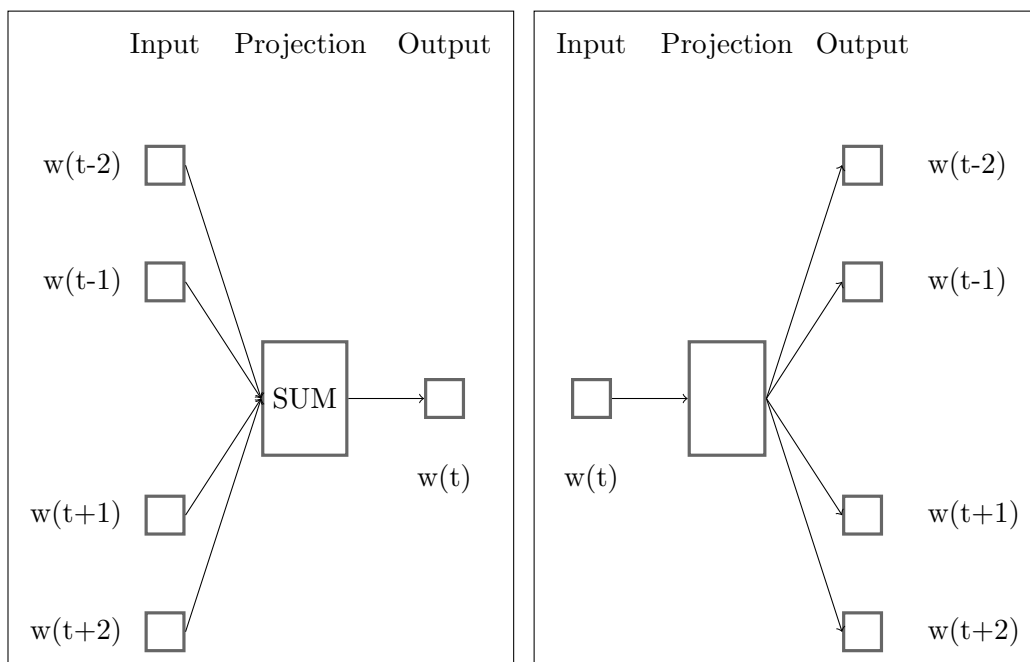

Figure 2.1: CBOW and Skip-gram schemes respectively

Due to its algorithmic simplicity and efficiency, Word2Vec has established itself as a strong baseline for numerous NLP tasks. Compared to more recent and complex models, Word2Vec requires minimal hyperparameter tuning, making it a relatively straightforward approach.

However, it is important to acknowledge that Word2Vec has limitations. These include its inability to capture **global information** within a document, its challenges in effectively handling **morphologically rich languages** (languages with many word variations), and its lack of awareness of the **broader context** beyond a limited window of surrounding words.

GloVe [14]

GloVe leverages the co-occurrence statistics of words within a corpus to learn vector representations. This approach involves constructing a co-occurrence matrix, where each entry reflects the frequency of two words appearing together within a predefined window size. This matrix essentially captures the relative importance of various word pairings.

A core principle of GloVe lies in the notion that word vectors should effectively encode the ratios between co-occurrence probabilities of words. By analyzing these ratios, GloVe can identify semantic relationships between words. This is achieved by factorizing the co-occurrence matrix into a lower-dimensional space, allowing for efficient representation and manipulation of word meanings.

To optimize the learned word embeddings, GloVe employs a weighted least squares objective function. This function aims to minimize the discrepancy between the dot product of two word vectors and the logarithm of their co-occurrence probability. Through iterative adjustments of the word vectors, GloVe converges on a solution that yields the desired word embeddings.

- **FastText** [13]

2.2 Transformer-based models

- Discuss traditional word embedding methods like FastText and their limitations.
- Explain the concept of transformer-based models like BERT and their advantages for text representation.
- Review related work on RAG algorithms and their dependence on effective text representations.Discuss existing research on evaluating text representations using analogy tests and confusion matrices.
- Briefly mention the UPV corpus set as the chosen evaluation benchmark.

■ 3 Methodology

■ 3.1 Evaluation process

This work specifically targets the evaluation of word, sentence, and paragraph representation methods on datasets in the Czech language. This focus on Czech allows for a deeper understanding of how these methods perform in a language with specific characteristics, such as a rich inflectional morphology and the presence of diacritics.

■ Text Data Preparation

In certain foreign languages, a common issue arises when individuals incorrectly write words by omitting diacritics or altering letters, Fig. 3.1. This problem is prevalent in social media, chatbots, and other informal written communications. As a result, embedding models face challenges in comprehending text without diacritics (hereinafter diacriticless). A potential solution involves adapting data representation to accommodate both formal and informal styles of writing.

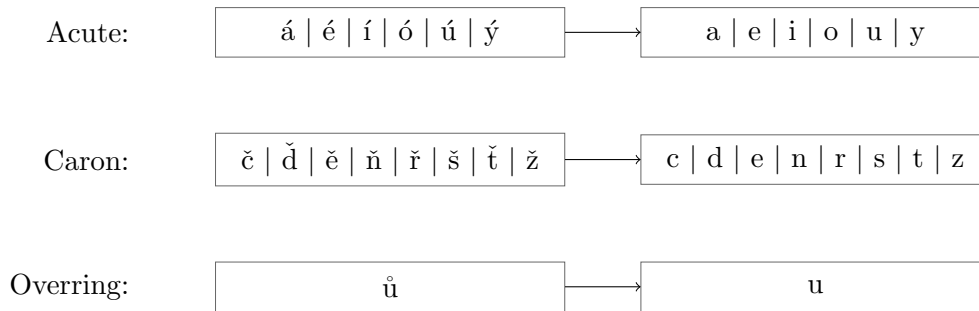| Acute:	á | é | í | ó | ú | ý	→	a | e | i | o | u | y
Caron:	č | ď | ě | ň | ř | š | ť | ž	→	c | d | e | n | r | s | t | z
Overring:	ů	→	u

Figure 3.1: Usual changes in informal czech texts.

This study will employ two distinct text representations: text with diacritics and text without diacritics. To ensure optimal evaluation, the diacritic text will be assessed using datasets that preserve these diacritics, while the diacriticless text will be evaluated using datasets that lack diacritics.

As detailed in Lst. 3.1, this script is used for creating diacriticless versions of the datasets.

```
sed 's/.*/\L&/' "$1" | iconv -f utf-8 -t ascii//TRANSLIT > diacriticless/"$1"
```

Listing 3.1: Script for removing diacritics using Unix utilities

■ Datasets

UPV FAQ

Dataset is comprised of frequently asked questions (FAQs) and their answers from the Industrial Property Office of the Czech Republic website, organized into four distinct groups, as detailed in Table 3.1.

This work will evaluate two key metrics using UPV FAQ dataset

- **Question matching accuracy**: This metric involves calculating the cosine similarity between all possible question pairs within a dataset. A question is considered successfully matched if its second-highest cosine similarity score corresponds to another question belonging to the same class (i.e., the question with the highest similarity is likely the same question itself). The overall question matching accuracy is then computed as the ratio of successfully matched questions to the total number of question pairs evaluated.
- **Answer matching accuracy**: This metric assesses the system's ability to identify the correct answer for a given question. The system accomplishes this by directly comparing the question with pre-generated answer embeddings. By evaluating the similarity between the question and each answer embedding, the system classifies the question as corresponding to the answer with the highest similarity score. The overall answer matching accuracy is then calculated as the proportion of questions for which the system correctly identifies the corresponding answer.

Tests set	Number of tests
FAQv5	2054
FAQ50	561
FAQ76	2025
FAQ76v2	1965
All	6605

Table 3.1: Information about UPV FAQ tests

■ **Baseline**

Due to the inherent morphological richness of the Czech language, this study adopts **FastText** as the baseline word embedding method. This decision is motivated by FastText's ability to effectively capture morphological variations within words, a characteristic that has been shown to be advantageous for languages like Czech. While other techniques like Word2Vec and GloVe have been explored for word embedding generation, they have demonstrated lower performance in this context. The FastText word embedding model will be trained using the fastText.cc library [1]

Training parameters

- Architecture: CBOW
- Vector dimensionality: 300
- Loss function: Negative sampling loss
- Dictionary threshold (the frequency of the word to be included in the dictionary): 130

Training data

FastText model is trained on a segment of a preprocessed Common Crawl repository, which encompasses raw data from web pages, as well as metadata and text extractions. Given the nature of this dataset, it is expected to include misspellings and text lacking diacritics.

---

[1]fastText.cc library website: https://fasttext.cc/

   (i) Eliminating duplicate entries
   (ii) Filtering out lines containing fewer than 9 characters
  (iii) Excluding URLs
  (iv) Breaking lines into individual sentences
   (v) Converting all text to lowercase
  (vi) Removing lines containing words exceeding 30 characters
 (vii) Excluding HTML tags with less than 100 characters
(viii) Removing lines surpassing 500 characters
  (ix) Omitting words longer than 21 characters

Processed corpus contains 3.87 billion words.

Model summary

The FastText models trained in this study exhibit differences in vocabulary size. The diacritic model possesses a dictionary of approximately 800,000 words, while the diacriticless model contains roughly 762,000 words. This discrepancy reflects the inherent reduction in word count due to the removal of diacritics in the diacriticless dataset.

These vocabulary sizes directly influence the number of trainable parameters within each model. The number of parameters (N) can be calculated using the formula (3.1)

$$N = V_{\text{size}}d + d \tag{3.1}$$

where V_{size} is model vocabulary size

       d is chosen dimensionality of word embeddings

Applying this formula to the vocabulary sizes of our models:

- The diacritics model possesses approximately 240 million parameters.
- The diacriticless model has approximately 229 million parameters.

Chosen transformer models

To ensure a comprehensive evaluation, a curated selection of embedding models will be utilized. This selection encompasses three distinct categories:

  (i) **Existing Czech Embedding Models**: This category incorporates established Czech embedding models developed within the Czech NLP community. Their inclusion allows for a focused analysis of how these models perform specifically for the Czech language.

 (ii) **Multilingual Models from the Massive Text Embedding Benchmark (MTEB)**: The evaluation will leverage highly regarded multilingual models readily available through the MTEB. This inclusion enables an assessment of how these models generalize to the Czech language, providing insights into their adaptability across languages.

(iii) **Popular Monolingual Models from the MTEB (rank is lower than 100)**: In addition to multilingual models, this selection will also include well-regarded monolingual models (models trained on a single language) from the MTEB. This allows for a comparative analysis of how these models, potentially trained on English or other high-resource languages, perform on the Czech dataset.

All models should be open-source. And sentence-bert. ¡1B. With information about training process. If exisit two version of models - monolingual (without czech) and multilingula, we evaluate only multilingual

mBERT [11]

Multilingual Bidirectional Encoder Representations from Transformers (mBERT) is a BERT model, trained on a Wikipedia dump of 100 languages. The model performs best on high-resource languages such as English, French and Chinese, since lower-resource languages are underrepresented in the training data.

Czert-B [7]

The Czert model is a set of Czech BERT-like language representation models developed specifically to enhance performance in processing the Czech language. These models leverage the BERT and A Lite BERT (ALBERT) [9] architectures and are designed to outperform multilingual models by training exclusively on Czech data. The training set includes a comprehensive corpus of Czech texts, such as Wikipedia articles, news, and other texts, accumulating to around 36GB of data.

There are 2 variants of the Czert model, Czert-A and Czert-B. Unfortunately Czert-A model is not available, so we will test only Czert-B model. Czert-B model is based on the traditional BERT architecture (110M parameters). Model are pre-trained from scratch using the Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks. However, a slight modification is made to the NSP task to adapt it better to the Czech language corpus structure.

Seznam's models [2]

This study leverages a group of compact word embedding models specifically designed for the Czech language. These models were developed by the Seznam research group with a focus on efficient word representation generation.

- RetroMAE-Small: BERT-small model trained with the Retrieval-oriented Language Models Via Masked Auto-Encoder (RetroMAE) objective [5] on a custom Czech corpus.
- Dist-MPNet-ParaCrawl: BERT-small model distilled from the *sentence-transformers/all-mpnet-base-v2* [2] model, using parallel cs-en dataset [12] for training.
- Dist-MPNet-CzEng: BERT-small model distilled from the *sentence-transformers/all-mpnet-base-v2* [3] model, using parallel cs-en dataset Czeng [8] for training.
- SimCSE-RetroMAE-Small: The RetroMAE-Small model fine-tuned with the Simple Contrastive Learning of Sentence Embeddings (SimCSE) [6].
- SimCSE-Dist-MPNet-ParaCrawl: The Dist-MPNet-ParaCrawl model fine-tuned with the SimCSE.
- SimCSE-Dist-MPNet-CzEng: The Dist-MPNet-CzEng fine-tuned with the SimCSE.
- SimCSE-Small-E-Czech: Czech ELECTRA model [0] fine-tuned with the SimCSE objective to enhance sentence embeddings.

---

[2]Model on the huggingface website: https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[3]Model on the huggingface website: https://huggingface.co/sentence-transformers/all-mpnet-base-v2

The developed models are about eight times smaller and five times faster than conventional base-sized models, making them suitable for real-time applications where computational efficiency is critical. The models are trained using techniques like pre-training, knowledge distillation, and unsupervised contrastive fine-tuning to adapt to the limited availability of labeled Czech data.

Multilingual E5 (mE5) [1]

distiluse-base-multilingual-cased-v2

paraphrase-multilingual-MiniLM-L12-v2

paraphrase-multilingual-mpnet-base-v2

LaBSE [4]

snowflake-arctic-embed

jina-embeddings

Only jina-embeddings-v2, because it's better in **mteb! (mteb!)**

nomic-embed-text

nomic-embed-text-v1 nomic-embed-text-v1.5

gte

gte-large-en-v1.5 gte-base-en-v1.5 gte-large gte-base gte-small gte-tiny

bge

bge-large-en-v1.5 bge-base-en-v1.5 bge-small-en-v1.5 bge-micro-v2

- **mxbai-embed-large-v1**

xlm-roberta [10]

XLM-RoBERTa is a multilingual version of RoBERTa. It is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.

UAE-Large-V1 [3]

- Describe the evaluation process for different text representations.
  - Specify the chosen word, sentence, and paragraph representation models (e.g., Fast-Text, BERT variants).
  - Explain the usage of analogy tests and confusion matrices for evaluation.

- - Detail the selection process for the UPV corpus set and its suitability for technical QA tasks.
- Outline the second part of the study focusing on RAG for technical QA.
  - - Explain the RAG algorithm and its reliance on text representations.
    - Describe the evaluation approach for selecting optimal representations for RAG.
    - Mention the factors considered during evaluation, such as embedding efficiency, text chunk size, and factuality of generated answers.

4 Experiments and Results

- Present the results of the evaluation for different text representations using analogy tests and confusion matrices.
- Discuss the findings regarding the effectiveness of each representation model for capturing semantic relationships in technical text.
- Analyze the results from the RAG evaluation, highlighting the impact of different representations and text chunk sizes on answer generation quality and CPU efficiency.
- Identify the representation model that achieves a balance between factuality of answers and computational demands.

◼ 5 Discussion

- Interpret the overall findings and their implications for choosing suitable text representations for RAG in technical QA tasks.
- Discuss the strengths and limitations of the chosen evaluation methods.
- Address potential challenges encountered during the study and suggest improvements for future research.

◼ 6 Conclusion

Summarize the achieved results. Can be similar as an abstract or an introduction, however, it should be written in past tense.

- Summarize the key takeaways from the research, emphasizing the most effective text representation model for RAG in technical QA based on the evaluation criteria.
- Briefly mention the trade-offs between factuality, CPU usage, and other factors in selecting representations for RAG.
- Suggest potential future research directions, such as exploring other text representation methods or evaluating RAG performance on different datasets.

■ 7   References

[1]   L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, *Multilingual e5 text embeddings: A technical report*, 2024. arXiv: 2402.05672 [cs.CL].

[2]   J. Bednář, J. Náplava, P. Barančíková, and O. Lisický, *Some like it small: Czech semantic embedding models for industry applications*, 2023. arXiv: 2311.13921 [cs.CL].

[3]   X. Li and J. Li, "Angle-optimized text embeddings," *arXiv preprint arXiv:2309.12871*, 2023.

[4]   F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, *Language-agnostic bert sentence embedding*, 2022. arXiv: 2007.01852 [cs.CL].

[5]   S. Xiao, Z. Liu, Y. Shao, and Z. Cao, *Retromae: Pre-training retrieval-oriented language models via masked auto-encoder*, 2022. arXiv: 2205.12035 [cs.CL].

[6]   T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6894–6910.

[0]   M. Kocián, J. Náplava, D. Štancl, and V. Kadlec, *Siamese bert-based model for web search relevance ranking evaluated on a new czech dataset*, 2021. arXiv: 2112.01810 [cs.IR].

[7]   J. Sido, O. Prazák, P. Pribáň, J. Pasek, M. Seják, and M. Konopík, "Czert - czech bert-like model for language representation," *CoRR*, vol. abs/2103.13031, 2021. arXiv: 2103.13031. [Online]. Available: https://arxiv.org/abs/2103.13031.

[8]   T. Kocmi, M. Popel, and O. Bojar, *Announcing czeng 2.0 parallel corpus with over 2 gigawords*, 2020. arXiv: 2007.03006 [cs.CL].

[9]   Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *Albert: A lite bert for self-supervised learning of language representations*, 2020. arXiv: 1909.11942 [cs.CL].

[10] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised cross-lingual representation learning at scale," *CoRR*, vol. abs/1911.02116, 2019. arXiv: 1911.02116. [Online]. Available: http://arxiv.org/abs/1911.02116.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[12] M. Esplà, M. Forcada, G. Ramírez-Sánchez, and H. Hoang, "Paracrawl: Web-scale parallel corpora for the languages of the eu," in *Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks*, European Association for Machine Translation, 2019, pp. 118–119.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

A Appendix A