Czech
Technical University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Automated Fact Checking Based on Czech Wikipedia

Bachelor's Thesis of
Tomáš Mlynář

Supervisor: **Ing. Herbert Ullrich**
Study program: **Open Informatics**
Specialization: **Artificial Intelligence and Computer Science**
May 2023

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Mlynář Tomáš** Personal ID number: **498831**

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:

Automated Fact Checking Based on Czech Wikipedia

Bachelor’s thesis title in Czech:

Automatizované ověřování faktů daty z české Wikipedie

Guidelines:

The overall task of this thesis is to combine methods for automated fact-checking recently developed at AIC in order to build a showcase application of the algorithms working on a snapshot of Czech Wikipedia.
1) Explore state-of-the-art machine learning methods dealing with the tasks of Document Retrieval and Natural Language Inference in Czech language.
2) Acquire Czech data for Wikipedia-based fact checking, possibly reusing or reproducing the results of [3].
3) Train appropriate models for the Document Retrieval and Natural Language Inference tasks on the resulting data.
4) Integrate the solutions into an initial version of the fact-checking pipeline.
5) Evaluate the selected models and their pipeline using standard methods.
6) Build a prototype showcase application for the resulting fact-checking system.

Bibliography / sources:

[1] Thorne, James, et al. "FEVER: a large-scale dataset for fact extraction and verification." arXiv preprint arXiv:1803.05355 (2018).
[2] Thorne, James, et al. "The fact extraction and verification (fever) shared task." arXiv preprint arXiv:1811.10971 (2018).
[3] Ullrich, Herbert "Dataset for Automated Fact Checking in Czech Language." CTU Master thesis, https://dspace.cvut.cz/handle/10467/95430 (2021).
[4] Rýpar, Martin "Methods of Document Retrieval for Fact Checking." CTU Master Thesis, https://dspace.cvut.cz/handle/10467/95315 (2021).

Name and workplace of bachelor’s thesis supervisor:

Ing. Herbert Ullrich Department of Computer Science FEE

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

Date of bachelor’s thesis assignment: **18.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

Ing. Herbert Ullrich	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would like to thank my supervisor Herbert Ullrich for his kind approach, valuable feedback and assistance during my work. I am also thankful to the head of the NLP research group at AIC, Jan Drchal, who provided helpful comments and expertise during the group's weekly meetings.

Additionally, I am also grateful to Kamila Etchegoyen Rosolová and Jana Boušová from the Center for Academic Writing of the Czech Academy of Sciences for their valuable proofreading, guidance and feedback.

Last but not least, I would like to mention my family and girlfriend for all their support throughout my bachelor studies.

The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged.

Declaration

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

In Prague, 26. May 2023

Abstract

Automated Czech fact-checking assists journalists in verifying claims when combating the spread of misinformation. This thesis builds upon previous research conducted at AIC, presents an updated localisation of the FEVER dataset and introduces and evaluates the NLI filtering approach for reducing noise in localised datasets. Moreover, I evaluated document retrieval methods and trained new natural language inference models on the filtered datasets. I integrated the NLI and document retrieval models into an initial version of the fact-checking pipeline and created a showcase application. The new dataset was localised, partially reusing previous works with new translations and processing. I compared instances of the NLI filtering using a fixed 0.7 threshold and thresholds maximising its F1 score and precision on annotated data. As for the document retrieval methods, I evaluated sparse and hybrid methods, producing a more robust hybrid Anserini+CrossEncoder model baseline. The NLI models were finetuned based on XLM-RoBERTa-large. Although the NLI filtering does decrease the percentage of noise in the annotated sample, the performance of the fine-tuned models does not significantly increase and, in some cases, even decreases. This drop in performance could be caused by the filtering model eliminating the challenging data points. The pipeline evaluation showed results comparable to previous works. The showcase application was developed using the Streamlit framework and enhanced with temperature scaling calibration, SHAP explainability, and new output modes for improved usability.

Keywords: Czech Wikipedia, Document Retrieval, Fact-checking, Fact-checking application, Fact-checking pipeline, Natural Language Inference

Supervisor: Ing. Herbert Ullrich

Abstrakt

Automatické ověřování faktů v češtině pomáhá novinářům ověřovat tvrzení v boji proti šíření dezinformací. Tato práce navazuje na předchozí výzkum provedený v AIC, představuje aktualizovanou lokalizaci datasetu FEVER a zavádí a vyhodnocuje NLI filtrování pro redukci šumu v lokalizovaných datasetech. Kromě toho jsem vyhodnotil metody vyhledávání dokumentů a natrénoval nové inferenční modely přirozeného jazyka s filtrovanými datasety. NLI modely a vyhledávání dokumentů jsem začlenil do počáteční verze pipeline pro ověřování faktů a vytvořil ukázkovou aplikaci. Nová datová sada byla lokalizována, přičemž byly částečně znovu použity předchozí práce s novými překlady a zpracováním. Porovnal jsem filtrování na základě fixní prahové hodnoty 0.7 a prahových hodnot maximalizujících F1 skóre a precision na anotovaných datech. Pokud jde o metody vyhledávání dokumentů, hodnotil jsem sparse a hybridní metodu, z nichž vyplynula baseline v podobě hybridní metody Anserini+CrossEncoder. Modely NLI byly finetunovány na základě XLM-RoBERTa-large. Přestože NLI filtrování zlepšuje transduction precision datasetů, výkonnost finetunovaných modelů se výrazně nezvyšuje a v některých případech dokonce klesá. Tento pokles výkonu může být způsoben tím, že filtrovací model eliminuje náročné datové body. Vyhodnocení pipeline ukázalo výsledky srovnatelné s předchozími pracemi. Předváděcí aplikace byla vyvinuta pomocí frameworku Streamlit a rozšířena o kalibraci pomocí temperature scaling, vysvětlitelnost pomocí SHAP a nové výstupní režimy pro lepší použitelnost.

Klíčová slova: Aplikace na ověřování faktů, Česká Wikipedie, Inference v přirozeném jazyce, Ověřování faktů, Pipeline na ověřování faktů, Vyhledávání dokumentů

Contents

Figures

Tables

Chapter 1
Introduction

> *Machines may get better at "mimicking human meaning," and thereby better at predicting human behavior, but "there's a difference between mimicking and reflecting meaning and originating meaning," Ferrucci said. That's a space human judgment will always occupy.*
>
> Philip Tetlock (2019)
> David Ferrucci

In today's world, many people are connected through the internet and use it daily to gather information. However, not all of this information is factually correct, leading to the spread of intentionally wrong information called misinformation and disinformation. The Czech Republic is not an exception. In the last few years, the number of articles published on disinformation sites has risen gradually from 167716 articles in 2019 to 197177 articles in 2021 (Česko v datech, 2022).

This spread of misinformation is one of the reasons why some journalists around the globe started to fact-check claims, whole newspaper articles, videos and photos. In the Czech Republic, the major fact-checking websites are *Demagog*[1], *Faktické Info*[2] (formerly part of Manipulátoři[3]), *Ověřovna!*[4] and *AFP Na pravou míru*[5]. Their work is difficult and time-consuming nowadays as they must manually find evidence for each claim and verify the claims.

To help these fact-checkers, the fact-check team at the Artificial Intelligence Center[6], led by Jan Drchal, started developing methods for automated fact-checking using state-of-the-art natural language processing (NLP) methods. The work of the fact-check team that was partially used in this thesis was done by Dědková (2021), Rýpar (2021), Ullrich (2021) and Ullrich et al. (2023).

The process of automated fact-checking can be divided into two main parts. The document retrieval part retrieves the top k matching evidence documents for a given claim. The second part is Natural Language Inference (NLI) also referred to as Recognising Textual Entailment (RTE) in some older works. In NLI, the model tells us whether the inference relation between two texts is entailment, contradiction or neutral (MacCartney et al., 2008). Later in this work these relations are labeled **SUPPORTS**, **REFUTES** and **NOT ENOUGH INFO (NEI)**.

To train new neural models for methods for automated fact-checking, it was necessary

[1] `https://demagog.cz/`
[2] `https://www.fakticke.info/`
[3] `https://manipulatori.cz/`
[4] `https://www.irozhlas.cz/zpravy-tag/overovna`
[5] `https://napravoumiru.afp.com/list`
[6] `https://www.aic.fel.cvut.cz/`

to acquire new Czech datasets. These datasets were based on data from Czech Wikipedia and from ČTK[7]. However, the Wikipedia dataset was imperfect because of the procedure used to obtain it (Ullrich et al., 2023). During this procedure, the original claims from English dataset FEVER, made by Thorne; Vlachos; Christodoulopoulos, et al. (2018), were translated and linked with evidence from Czech Wikipedia. Because the information in the Czech and English language mutations of Wikipedia was not perfectly aligned, the resulting dataset was noisy. To eliminate the problem with noise, Ullrich et al. (2023) proposed a method to filter the data using a state-of-the-art NLI model and acquire a new cleaner dataset.

The main contributions of this bachelor thesis are the filtering approach using an NLI model and a discussion of its performance, a new dataset based on the Czech Wikipedia, and a prototype showcase application developed to present the functions of the fact-checking pipeline. Based on the new datasets, NLI and document retrieval models were trained, integrated into the initial version of the fact-checking pipeline based on the pipeline developed by Ullrich et al. (2023), and evaluated. The newly acquired pipeline could serve as a baseline for other fact-checking tools in Czech language and low-resource languages in the future.

1.1 Thesis Outline

This section provides a succinct description of each chapter in this thesis.

- **Chapter 1** introduces the thesis, its main goals and some background.
- **Chapter 2** describes the state-of-the methods in NLI, document retrieval, and the used metrics.
- **Chapter 3** describes the dataset creation and preparation. Moreover, it introduces the NLI filtering approach.
- **Chapter 4** describes the training of NLI models and the preparation of document retrieval methods. It also evaluates them.
- **Chapter 5** describes the whole fact-checking pipeline and some additional features.
- **Chapter 6** introduces a showcase application for the pipeline.
- **Chapter 7** concludes this thesis.

[7] Česká tisková kancelář - Czech News Agency.

Chapter 2
State-of-the-Art Overview

This chapter describes current state-of-the-art methods used in the Czech automated fact-checking task. Automated fact-checking, as described by Thorne; Vlachos; Cocarascu, et al. (2018), consists of several stages. In this thesis, two main stages are considered: **document retrieval** and **natural language inference (NLI)**. These stages are later combined into one more complex pipeline, which is described in chapter 5. Therefore, to describe the state-of-the-art methods, it is possible to focus on each stage separately, in section 2.1 on document retrieval and in section 2.2 on NLI. Moreover, because of the public boom of commercial services based on Large Pretrained Language Models (LPLMs) early in 2023, section 2.3 describing their fact-checking abilities in Czech language is added. Last section 2.4 provides an overview of used metrics.

2.1 Document Retrieval in Czech Language

Document retrieval is a task in NLP where a model retrieves a desired number of documents from a collection (corpus) called the *knowledge base* most relevant to the entered query. The document can vary in length from a whole Wikipedia page through paragraph-long texts to single sentences. Sometimes it is also called *Text retrieval*, which can be recognised as a part of *Information retrieval*. *"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)"* (Manning et al., 2008). In the fact-check team, Czech document retrieval was investigated in the works of Rýpar (2021) and Dědková (2021). According to Luan et al. (2021), document retrieval approaches can be divided into *sparse*, *dense* and *sparse-dense hybrids*. This division is used in the following subsections to describe state-of-the-art document retrieval methods.

2.1.1 Sparse Approach

The sparse approach (in the work of Rýpar (2021) called traditional) is an approach in document retrieval where each document is encoded to a sparse vector. The sparse vector used in bag-of-words models like TF-IDF and BM25 is usually a vector in $\mathbb{R}^v$ where v is the vocabulary size and the relevance score is taken as the inner product between document d and query q vectors $\langle q, d \rangle$ (Luan et al., 2021). The term sparse means that most of the vector components are zero because the bag-of-words methods encode sequences to vectors of occurrences of each word from the vocabulary. Below is a brief description of two representatives of this category chosen by Rýpar (2021) as baselines.

TF-IDF (DrQA)

TF-IDF is one of the traditional yet still effective methods for document retrieval task. The acronym TF-IDF stands for *Term Frequency - Inverse Document Frequency*. Its functionality is based on these two terms and computes the weight of a term t in a document d by the following formula:

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t.$$

The $tf_{t,d}$ is a term frequency of the term t in document d. The idf_t means inverse document frequency of the term t and is computed by the following formula:

$$idf_t = log\frac{N}{df_t}$$

where N is the number of all documents in the collection and df_t is the number of documents in the collection that contains the term t. According to these weights, a sparse vector of $tf\text{-}idf$ weights is then initialized (Manning et al., 2008).

One of the well-known implementations of the TF-IDF used by Rýpar (2021) and Dědková (2021) is an implementation from the DrQA system made by Chen et al. (2017).

BM25 (Anserini)

Another approach to assigning weights was introduced by Robertson et al. (2009). This approach was BM25 which stands for Best Match 25 and is also called Okapi weighting. According to Manning et al. (2008), it can be expressed as the following simplified formula for short queries:

$$BM25_{t,d} = \log idf_t \cdot \frac{(k_1+1)tf_{t,d}}{k_1((1-b)+b(L_d/L_{avg}))+tf_{t,d}}.$$

The main advantage is introducing new tuning parameters k_1 and b, where k_1 calibrates document term frequency scaling and b influences scaling by the document length (L_d means the length of document d and L_{avg} means the average length of all documents in the collection). Overall, that means that k_1 influences a level of saturation in term frequency, after which the score did not rise so steeply as in TF-IDF, and b tells us how much we want the score to be influenced by the document length (Manning et al., 2008).

The frequently used implementation of BM25 is in the Anserini library, which also has a Python interface called Pyserini developed by Lin et al. (2021). This implementation was also used in the works of the fact-check team members Dědková (2021) and Rýpar (2021).

2.1.2 Dense Approach

In contrast with the sparse approach, the dense approach nowadays encodes documents and queries as dense vectors using language models based on the Transformer architecture. Some state-of-the-art models were made using the dense approach. However, according to Luan et al. (2021), they are unfeasible for large-scale document retrieval, because of their computational demands. Dense retrieval can be divided into some paradigms. Most of them are depicted in figure 2.1.

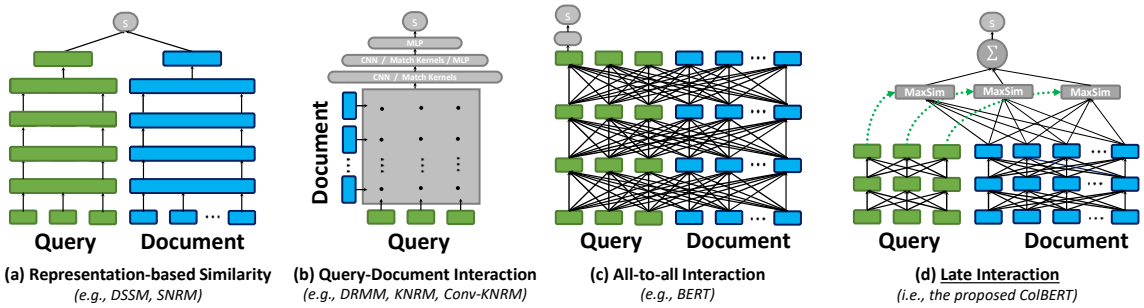


Figure 2.1: Diagrams of different paradigms used in dense retrieval. (reprinted from Khattab et al. (2020))

Two-Tower Paradigm

The two-tower paradigm (representation-based) is the most similar to the sparse retrieval. However, here are the vectors predicted by neural models. The vectors for documents can be precomputed before retrieval. During retrieval, only the query vector is computed and compared to the documents' vectors. Therefore, this paradigm is computationally more efficient during retrieval (Khattab et al., 2020). It is displayed in subfigure (a) in figure 2.1.

For Czech retrieval, Rýpar (2021) trained mBERT, which performed well, especially on the Wikipedia-based dataset CsFEVER[1], where it was the best-performing retrieval solution.

All-to-All Paradigm

All-to-all interaction paradigm (in the work of Rýpar (2021) called *Cross-attention*) enables interaction between words in the query, in the document and between both. This paradigm is depicted in subfigure (c) in figure 2.1. Khattab et al. (2020) state that this paradigm provides superior results. However, it is computationally not feasible for large-scale document retrieval because the computations must be made within and across query and document for every such pair during the retrieval (nothing can be precomputed).

The Query-Document interaction paradigm illustrated in subfigure (b) in figure 2.1 has similar disadvantages as the all-to-all paradigm. Moreover, it nowadays does not achieve state-of-the-art results.

Late Interaction Paradigm

To balance the quality, which is the domain of the *Cross-attention* paradigm and the cost of document retrieval, which is better in the *Two-tower* paradigm, Khattab et al. (2020) introduced **ColBERT**. ColBERT is a model based on Contextualized late interaction paradigm over BERT. In this paradigm, which can be seen in subfigure (d) in figure 2.1, representations of documents can be precomputed offline, and thus the cost of retrieval is reduced. Those document representations and representations of queries are encoded by 2 BERT models into a set of vectors and then compared by a maximum similarity measure, which is cosine similarity between a query and document encodings. Because of the maximum similarity, it provides better results than results achieved by using the *Two-tower* paradigm (Khattab et al., 2020).

[1]The same dataset is across works named *FEVER CS*, *CS FEVER* and *CsFEVER*. The name *CsFEVER* was chosen to refer to this dataset in this thesis because it is used in the latest paper (Ullrich et al., 2023).

The only disadvantage of this model is its extensive memory usage, which has been shown to cause performance problems in practice. To mitigate this problem, Santhanam et al. (2022) introduced a more efficient **ColBERTv2**, which, as they stated, outperforms existing retrievers. The improvement was achieved by compressing the representations of documents by clustering them into centroids and saving approximate residual representations.

The ColBERT model was also used in the work of Rýpar (2021), where it achieved the best results on ČTK based dataset among tested methods of document retrieval for Czech fact-checking.

SEAL

Search Engines with Autoregressive LMs (**SEAL**) is a document retrieval solution introduced by Bevilacqua et al. (2022). It shows promising results that match or outperform recent retrieval solutions but require less memory. Its function is based on an autoregressive language model combined with FM-index. During the retrieval, the autoregressive language model is used to generate multiple ngrams. These ngrams are then used to find a document from the collection (Bevilacqua et al., 2022).

Figure 2.2 describes high-level SEAL architecture. *"High-level SEAL architecture, composed of an autoregressive LM paired with an FM-Index, for which we show the first (F) and last (L) columns of the underlying matrix (more details in Sec 3.1). The FM-index constraints the autoregressive generation (e.g., after carbon the model is constrained to generate either tax, dioxide or atom in the example) and provides the documents matching (i.e., containing) the generated ngram (at each decoding step)"* (Bevilacqua et al., 2022).

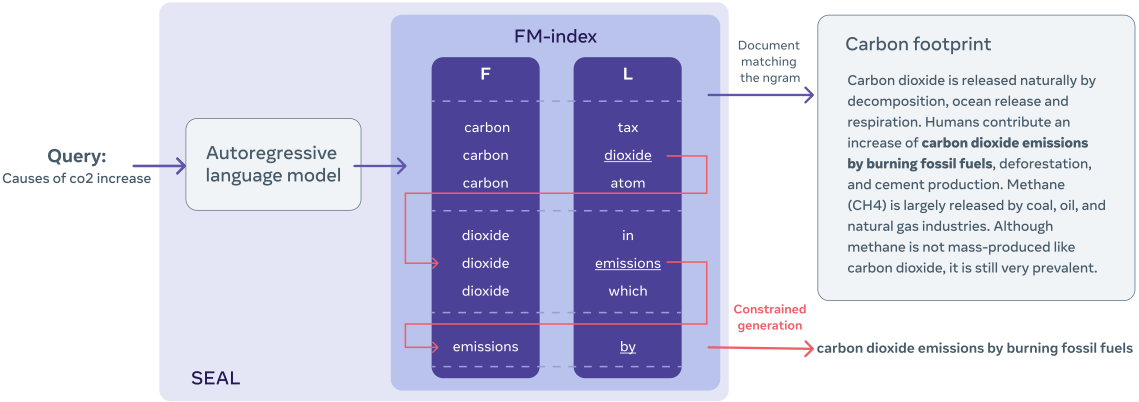


Figure 2.2: High-level SEAL architecture. (reprinted from Bevilacqua et al. (2022))

2.1.3 Hybrid Approach

It was shown that to achieve state-of-the-art results on large-scale document retrieval, it is possible to combine sparse and dense retrieval approaches to a sparse-dense hybrid approach. This approach was tested by Luan et al. (2021) and Qu et al. (2021) and shows promising results. In the fact-check team, this approach was tested for Czech fact-checking purposes in the work of Dědková (2021).

This approach usually consists of two phases. The first phase by using one of the sparse methods mentioned in subsection 2.1.1 preselects a smaller subset of all documents. Then this subset is passed to the second phase, where the subset is reranked by a dense

retriever, specifically usually with a retriever from the *All-to-all interaction* paradigm. Dědková (2021) calls these phases *document preselection* and *document reranking*, respectively, and these designations will are used later in this thesis. This process is illustrated in diagram 2.3.

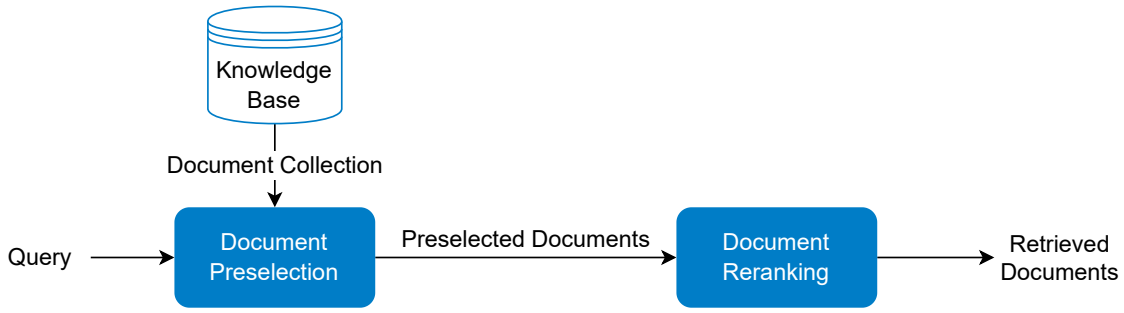


Figure 2.3: Hybrid Retrieval Diagram.

Specifically, the work of Luan et al. (2021) achieves promising results combining BM25 and BERT-based models. A similar setup was also used in the work of Dědková (2021) where the best solution for CsFEVER was the combination of BM25 implementation in Anserini by Lin et al. (2021) and fine-tuned DistilBERT[2].

2.2 Natural Language Inference in Czech Language

As described in the introduction, Natural language inference (NLI) is a task of NLP. In this task, the NLI model tells us whether the inference relation between two texts is entailment, contradiction or neutral (MacCartney et al., 2008). These two texts can vary in length.

In a fact-checking task, one of these texts is called a *claim*, and the second is called *evidence*. The claim is usually one sentence or phrase that should be fact-checked. Evidence is then a document retrieved by one of the document retrieval models, and its properties are described in section 2.1.

This thesis labels the inference relations as follows: **SUPPORTS** for entailment, **REFUTES** for contradiction and **NOT ENOUGH INFO (NEI)** for neutral relation. These labels were chosen to comply with labels from the work of Ullrich et al. (2023).

NLI is nowadays done with models using the Transformer architecture (Vaswani et al., 2017). For Czech NLI, it is necessary to fine-tune models that were pre-trained on multilingual data (including Czech data), such as *mBERT* (Devlin et al., 2019), *Slavic BERT* (Arkhipov et al., 2019) or *XLM-RoBERTa* (Conneau et al., 2020) or Czech models like *CZERT* (Sido et al., 2021) or *RobeCzech* (Straka et al., 2021). According to the experiments in the work of Ullrich et al. (2023), the best-performing model is *XLM-RoBERTa* and, therefore, it is briefly described below.

2.2.1 XLM-RoBERTa

XLM-RoBERTa is a multilingual version of the RoBERTa model. It was trained on 2.5 TB of ComonCrawl data in 100 languages (Conneau et al., 2020). RoBERTa is a model based on BERT with an improved training procedure[3] (Liu et al., 2019).

[2]Used version is *distilbert-base-nli-stsb-mean-tokens* from the Sentence Transformers library.

[3]For example from the two BERT training objectives: Masked Language Model (MLM) and Next Sentence Prediction (NSP), only MLM is used in RoBERTa.

As mentioned above, this model achieved state-of-the-art results in Czech NLI, according to the work of Ullrich et al. (2023). These results were achieved using previously fine-tuned versions of XLM-RoBERTa. One of these was made by Deepset[4] and fine-tuned on the SQuAD2 (Rajpurkar et al., 2018)[5] task, whereas the second was trained by Huggingface on the XNLI task (Conneau et al., 2020)[6]. These previously fine-tuned models were then again finetuned on the CsFEVER dataset.

2.3 Large Pretrained Language Models

As described in the essay by Manning (2022), Large Pretrained Language Models (LPLMs) are nowadays used even without the fine-tuning step. That means that these models can be used only by specifying the task within the input itself. Early in 2023, there was significant progress in this area after the spread of the ChatGPT service based on the GPT3 (Brown et al., 2020) model, originally released in November 2022 by OpenAI (2022). Other relevant LPLMs are LaMDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2022), LLaMA (Touvron et al., 2023) or the last introduced GPT4 (OpenAI, 2023).

For fact-checking it is necessary not only to decide about the inference, but also provide evidence supporting the claim so the human journalist can review the decision. The only service providing evidence for the answers is as of March 2023, the AI-powered Microsoft Bing described below.

A similar approach that uses only one language model was explored by Lee et al. (2020). Despite using only a smaller model based on BERT (Devlin et al., 2019), they achieved comparable results to standard baselines for the FEVER dataset (Thorne; Vlachos; Cocarascu, et al., 2018).

2.3.1 AI-powered Microsoft Bing

Early in 2023, Microsoft announced the launch of a new AI-powered Microsoft Bing[7] in a blog post by Mehdi (2023b). Later they said that this new AI-powered Bing uses a GPT4 model from OpenAI (Mehdi, 2023a). This new Bing has also, among many other abilities, the ability to fact-check the entered claims and provide evidence, which is necessary for the task of fact-checking.

First, I asked a simple claim: *"Miloš Zeman je prezidentem."*[8][9], and the provided answer to the claim is shown in figure 2.4 (retrieved evidence is 1. cs.wikipedia.org[10], 2. aktualne.cz[11]), 3. bing.com[12], 4. seznamzpravy.cz[13].)

Second experiment was done with a more complicated query: *"Ověř mi fakt následující po dvojtečce pouze s využitím informací z české Wikipedie: Miloš Zeman je prezidentem"*[14]

[4] https://www.deepset.ai/

[5] https://huggingface.co/deepset/xlm-roberta-large-squad2

[6] https://huggingface.co/joeddav/xlm-roberta-large-xnli

[7] https://www.bing.com/new

[8] I asked on 26. February 2023 (still during the presidency of Miloš Zeman).

[9] In English: *Miloš Zeman is the president.*

[10] https://cs.wikipedia.org/wiki/Milo%C5%A1_Zeman

[11] https://www.aktualne.cz/wiki/osobnosti/politici/milos-zeman/r~i:wiki:423/

[12] https://www.bing.com/search?q=Milo%C5%A1+Zeman+je+prezidentem&toWww=1&redig=0AAEC2D0F3F145758FEBB85D83842D48

[13] https://www.seznamzpravy.cz/tag/milos-zeman-19093

[14] In English: *Verify for me the following fact after the colon using only information from the Czech Wikipedia: Miloš Zeman is the president*

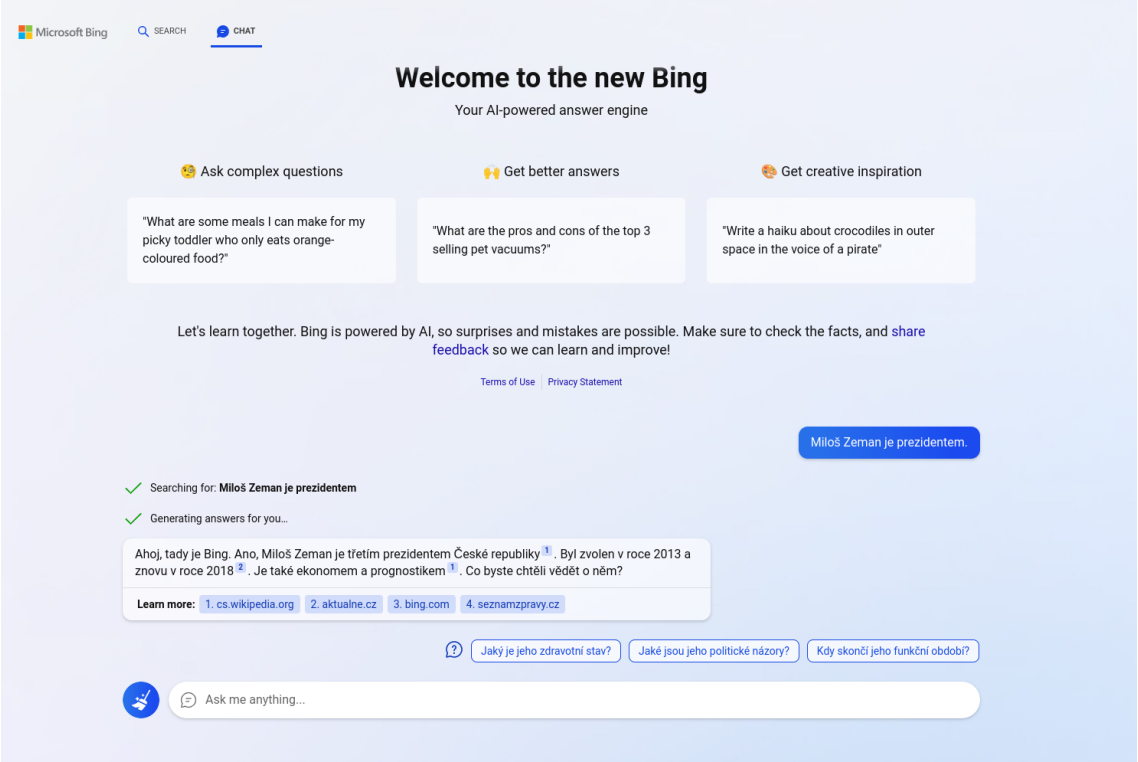


Figure 2.4: AI-powered Microsoft Bing.

to more precisely simulate the task of this thesis. The answer is shown in figure 2.5.

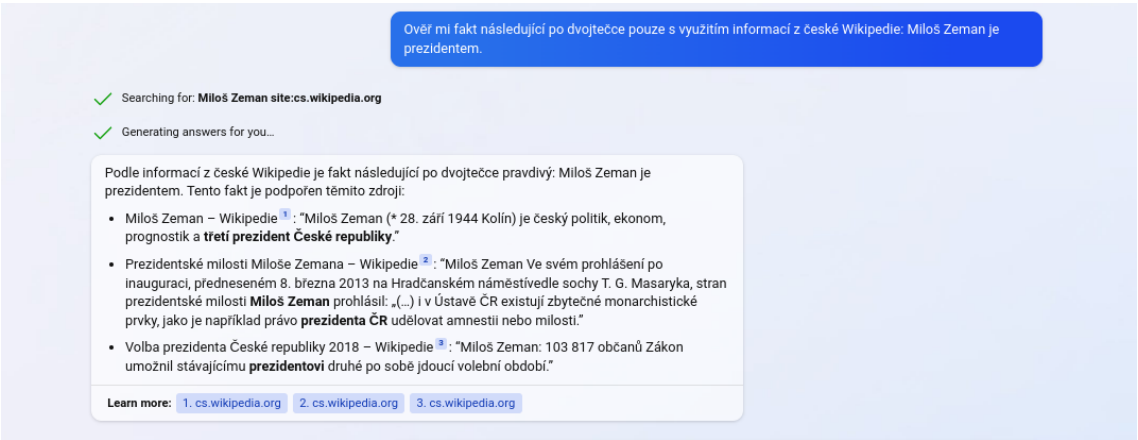


Figure 2.5: AI-powered Microsoft Bing - extended question.

In both cases, the model provided a correct answer and correct evidence. Therefore, I conclude that the performance of new LPLMs, especially Bing AI, should be investigated in future works to measure whether it provides state-of-the-art results. Overall, the future research subject will be the performance of LPLMs that are not available to run locally (or are not feasible to run locally) in NLP tasks in the Czech language.

2.4 Metrics

This section describes all primary metrics used in this thesis in one place.

- **Accuracy** is one of the standard statistical metrics. It is computed with the following formula:

$$Accuracy = \frac{|classifications|}{|all\ classifications|}.$$

- **Precision** is a standard metric in machine learning (ML) for two classes computed as

$$Precision = \frac{|True\ Positive|}{|True\ Positive| + |False\ Positive|}.$$

  It tells us what proportion of all predicted or retrieved elements are correctly identified.

- **Recall** is another standard metric in ML. It is computed as

$$Recall = \frac{|True\ Positive|}{|True\ Positive| + |False\ Negative|}$$

  and tells us the ratio between correctly predicted or retrieved elements and all elements that actually should be predicted or retrieved.

- **F-score** is a harmonic mean of precision and recall, representing them as one metric. The most used type of F-score is the F_1 **score**, where are precision and recall taken into account equally. It is computed as

$$F_1 = 2 * \frac{precision \cdot recall}{precision + recall}.$$

  F_1 score can be more generalized to the F_β **score** where a new parameter β is introduced that allows us to control the influence of recall and precision[15]. F_β score is computed with the following formula:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}.$$

- **Mean Reciprocal Rank (MRR)** is a standard metric used in document retrieval evaluation. It measures the quality of the retrieval by considering the rank of the first relevant answer. The formula used to compute MRR is

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i}$$

  where Q is the number of all tested queries and $rank_i$ is the rank of the first relevant information retrieved using the *ith* query. If no relevant information is retrieved, this query's term is set to 0.

[15] Recall is β times more important than precision.

Chapter 3
Data

This chapter describes acquiring Czech data for Wikipedia-based fact-checking. First, the related and previous works are discussed, then the whole solution is presented, and then each part is described more thoroughly in the following sections.

3.1 Related Works

It is necessary to acquire a corresponding dataset to train document retrieval and NLI models for the fact-checking pipeline. In the English language exists a FEVER dataset made by Thorne; Vlachos; Christodoulopoulos, et al. (2018). FEVER is a large-scale dataset for Fact Extraction and VERification, and it is suitable for English fact-checking. It is based on Wikipedia and has the following structure:

```
{
    "id": 139037,
    "verifiable": "VERIFIABLE",
    "label": "SUPPORTS",
    "claim": "Star Trek: Discovery is a series.",
    "evidence": [[[161821, 176242, "Star_Trek-COLON-_Discovery", 0]],
                 [[161821, 176243, "Star_Trek-COLON-_Discovery", 1]],
                 [[161821, 176244, "Star_Trek-COLON-_Discovery", 6], [16
    ↪ 1821, 176244, "Sonequa_Martin-Green", 3]]]
}
```

Each data point consists of an **id** that is unique in the dataset, information whether it is **verifiable** $\in$ {VERIFIABLE, NOT VERIFIABLE},
a **label** $\in$ {SUPPORTS, REFUTES, NOT ENOUGH INFO}, a **claim** sentence and a list of **evidence** sets where each has at least one evidence with an **Annotation ID** and **Evidence ID** for debugging purposes, **Wikipedia URL** and **Sentence ID**.

As this thesis is aimed at Czech fact-checking, it is necessary to acquire a Czech dataset. There are two main ways to achieve that. Human annotators can collect the dataset, like the CTKFacts dataset from Ullrich (2021). Alternatively, it can be localized from datasets in other languages like FEVER as the CsFEVER dataset from Ullrich (2021). The more economical approach is localization because it does not require human annotators. However, as stated by Ullrich et al. (2023), it yields noisy results because of the process used.

The noise emerges when, as described in (Ullrich, 2021), the Czech articles are linked to the evidence in claims. The Czech Wikipedia sometimes has different information about the topic. There is often no related information, so the data point should have the NEI

label instead of being verifiable. An example of such a case is shown in table 3.1. The second type of occurring noise is the change in the claim truthfulness in the time (e.g. the claim *"Santorini má 15 550 obyvatel."*).

Czech claim	Steven Zaillian získal v roce 2011 cenu od Writers Guild of America.
Original claim	Steven Zaillian won an award from the Writers Guild of America in 2011.
Czech evidence	Steven Zaillian (* 30. ledna 1953 Fresno) je americký scenárista a režisér. Studoval na Sanfranciské státní univerzitě. Je držitelem mnoha ocenění, včetně Zlatého glóbu a Oscara. Začínal koncem sedmdesátých let jako stříhač. Svůj režijní debut s názvem "Nevinné tahy" natočil v roce 1993. Později natočil ještě filmy "Žaloba" (1998) a "Všichni královi muži" (2006). V roce 2016 režíroval sedm z osmi epizod seriálu "Jedna noc". Jen jako scenárista se podílel například na snímcích "Schindlerův seznam" (1993), "Gangy New Yorku" (2002) a "Muži, kteří nenávidí ženy" (2011).
Part of original evidence	Steven Ernest Bernard Zaillian (born January 30, 1953) is an American screenwriter, film director and producer. He won an Academy Award, a Golden Globe Award and a BAFTA Award for his screenplay Schindler's List (1993) and has earned Oscar nominations for the films Awakenings, Gangs of New York, Moneyball and The Irishman. He was presented with the Distinguished Screenwriter Award at the 2009 Austin Film Festival and **the Laurel Award for Screenwriting Achievement from the Writers Guild of America in 2011**. Zaillian is the founder of Film Rites, a film production company. In 2016, he created, wrote and directed the HBO limited series The Night Of. ...

Table 3.1: Example of change in evidence due to localization

3.2 Proposed Solution

It was decided to partially reproduce and reuse the approach presented by Ullrich (2021) with some changes. The first change is that after a discussion with my supervisor Herbert Ullrich, we decided to maintain the splits of the original FEVER to prevent document leakage as described in (Ullrich, 2021) rather than merge the train and dev splits of the dataset and try to build balanced splits.[1] The second primary change is the NLI filtering approach (which will be described in section 3.8.2) which should handle the noise appearing after the localization phases as described in section 3.1. Other changes include using newer machine translation services and new libraries for data processing. That was done because the quality of available machine translators and libraries is now better than when

[1]The original FEVER test split is not available. Therefore, splits from (Thorne; Vlachos; Christodoulopoulos, et al., 2018) were used.

making CsFEVER in (Ullrich, 2021). All changes are described in the corresponding sections.

The whole process of localization of the dataset consists of the following steps:

1. Acquire a **Wikipedia dump** in the Czech language and clean it from incorrect white spaces. Optionally a dictionary mapping ids to revision ids of Wikipedia can be acquired. (section 3.3)
2. **Translate** the original FEVER claims using machine translation methods to the Czech language. (section 3.4)
3. **Map** Wikipedia articles names used in original FEVER evidence to Czech Wikipedia articles used in evidence of verifiable data points (section 3.5)
4. **Unfold** the evidence sets to be able to use NLI models correctly for filtering. (section 3.6)
5. Apply the **NLI filtering** method on verifiable data points. (section 3.8)

3.3 Wikipedia Dump

The first step is acquiring a Wikipedia dump. The dump was downloaded from the official site of Wikipedia.[2] Because of the length of work, it was necessary to fix the version of the Wikipedia dump to version 20220801 from 1. August 2022. After the download, the dump was extracted using the *WikiExtractor* library by Attardi (2015). Then the dump was processed to remove wrong white space symbols, such as newlines and non-breaking spaces. A simple Python Jupyter notebook providing all the functions and other experimental approaches (including using the Gensim library and extracting the first paragraphs[3]) was made and is in the enclosed repository.

To train good neural NLP models, it is also necessary to normalize the Unicode to one of the canonical normal forms (NFC, NFD) and fix it, which was done by the *ftfy* library (Speer, 2019). In this thesis, the Normalization Form C (NFC) was chosen because the provided Unicode is shorter than in Normalization Form D (NFD), where the canonical decomposition is not followed by canonical composition.[4]

Moreover, a simple local database to store the dump was made using SQLite to provide more straightforward access to the dump for later use in the showcase application and other scripts. A Jupyter notebook implementing this is enclosed in the repository.

3.4 Translation

Machine translation methods were used as the original FEVER claims must be translated into the Czech language. For the translation were selected 3 candidate machine translators:

- WMT21 En-X by Tran et al. (2021) (Meta AI)[5] (later refered to as Facebook).
- Google Translation API[6] (later refered to as Google).

[2] `https://dumps.wikimedia.org/cswiki/`

[3] This approach was not used because the extracting does not function reliably and the longer evidence should provide more information for the NLI model even when it is cropped by the maximum number of tokens.

[4] `http://www.unicode.org/reports/tr15/`

[5] `https://huggingface.co/facebook/wmt21-dense-24-wide-en-x`

[6] `https://cloud.google.com/translate`

- DeepL API[7] (later refered to as DeepL).

For using the Facebook model, it was necessary to download it from the repository and prepare it for running locally on the RCI cluster's GPU. Google and DeepL translators were used with the provided API. The implementations of all of the translators are in the enclosed repository, together with the corresponding Jupyter notebooks.

First, a small sample of 20 claims was selected. Ten of them from wrongly translated claims from the dataset made by Ullrich (2021) (supposed to be hard to translate), and ten were randomly selected from the FEVER dataset. After that, these claims were translated using all three translators and manually evaluated with a simple scoring method.[8] The simple scoring method was used because the standard metrics for the machine translation, such as BLEU (Papineni et al., 2002), need a large sample of human reference translations, which is not feasible in this thesis. A sample translation is shown in table 3.2, and all translations are attached in Appendix A.

Original FEVER claim	Congressional Space Medal of Honor is the highest award given only to astronauts by NASA.
Facebook	Congressional Space Medal of Honor je nejvyšší ocenění, které NASA uděluje pouze astronautům.
Google	Congressional Space Medal of Honor je nejvyšší ocenění, které NASA uděluje pouze astronautům.
Deepl	Vesmírná medaile cti Kongresu je nejvyšší ocenění, které NASA uděluje pouze astronautům.

Table 3.2: Translation example.

In this sample translation, we see that DeepL tends to provide a full translation. However, this example is wrongly translated as the correct Czech translation is *"Kongresová kosmická medaile cti"*[9]. Moreover, it is better to maintain the original name in this case because it can be used in the Czech language without translation. Therefore, the translations made by Facebook and Google are considered better because they align with Czech Wikipedia. As this happened in more cases, I decided to choose Google or the Facebook translator. When compared, they provided either the same translation or Facebook performed better.[10] Thus, I chose the Facebook model to translate all original FEVER claims.

3.5 Mapping to Czech Wikipedia

To provide correct evidence for the now-translated claims, it is necessary to link the Wikipedia pages from the evidence of each claim to its Czech version. To do that, the MediaWiki software provides an API for Wikipedia which returns corresponding URL links in other languages.[11] The implementation is located in the enclosed repository.

[7] `https://www.deepl.com/pro-api`
[8] 3 - Excellent, 2 - Good, 1 - Usable, 0 - Not Usable (changes context)
[9] According to `https://cs.wikipedia.org/wiki/Congressional_Space_Medal_of_Honor`.
[10] Final scores were: Google 25 points, DeepL 30 points and Facebook 36 points.
[11] `https://www.mediawiki.org/wiki/API:Langlinks`

3.6 Unfolding of Evidence

For the usage of the datasets with models, it is necessary to unfold the evidence sets because in the original FEVER there can be more evidence sets for one claim (shown in section 3.1). During the unfolding are from the original evidence consisting of n evidence sets created n new data points, each with one evidence set (all other fields of the data points are the same among the n data points, including the id number). This process is shown in figure 3.1.

Listing 3.1: Original data point.

```
{
    ...
    'evidence': [[['Aineiás', 'Aeneas'], ['Aeneis', 'Aeneid']], [['
    ↪ Aineiás', 'Aeneas']], [['Aineiás', 'Aeneas'], ['Ilias', 'Iliad']]]
    ...
}

```

Listing 3.2: 2 new data points.

```
{
    ...
    'evidence': [['Aineiás', 'Aeneas'], ['Aeneis', 'Aeneid']]
    ...
}
{
    ...
    'evidence': [['Aineiás', 'Aeneas']], [['Aineiás', 'Aeneas'], ['Ilias
    ↪ ', 'Iliad']]
    ...
}

```

Figure 3.1: Example of the process of unfolding the evidence sets.

3.7 Resulting Noisy Dataset

The distribution of the resulting noisy dataset made by the steps mentioned above is shown in table 3.4 for comparison with the distribution of labels in the original FEVER splits that are shown in table 3.3. The loss of data points is mainly caused by mapping in the step described in section 3.5, where some Wikipedia articles from evidence from the original FEVER are not available in Czech Wikipedia. A slight loss also occurred when the URL links were linked with evidence from the dump because some articles were not present in the dump for unknown reasons.

For later use for NLI filtering, the evidence sets were unfolded and the distribution of labels of the new dataset is shown in table 3.5. The dataset is available for use from a

	SUPPORTS	REFUTES	NEI
train	80035	29775	35639
dev	3333	3333	3333
test	3333	3333	3333

Table 3.3: FEVER dataset - distribution of labels.

	SUPPORTS	REFUTES	NEI	∑
train	55905	20792	35639	112336
dev	1957	1934	3333	7224
test	1993	1937	3333	7263
∑	59855	24663	42305	126823

Table 3.4: Noisy dataset - distribution of labels.

Huggingface repository.[12]

	SUPPORTS	REFUTES	NEI	∑
train	60661	22650	35639	118950
dev	2066	2059	3333	7458
test	2142	2045	3333	7520
∑	64869	26754	42305	133928

Table 3.5: Unfolded noisy dataset - distribution of labels.

3.8 Filtering

After the new noisy dataset based on Czech Wikipedia was obtained, the noise had to be filtered out. To do that, some other possible approaches were examined, as mentioned in the following subsection, but nothing from the papers was useful and adaptable for the filtering task in this thesis. The only other idea for filtering was mentioned by Ullrich et al. (2023), where is presented the idea that filtering using a finetuned NLI model could work. Therefore, this approach was selected and further examined in subsection 3.8.2.

3.8.1 Other Approaches

One of the possible approaches is using an Area Under the Margin (AUM) statistic for identifying mislabeled data (Pleiss et al., 2020). However, as Talukdar et al. (2021) state, it is not very promising in NLP because it also removes relevant information. Therefore, this approach was not further examined.

Another relevant work was done by Jeatrakul et al. (2010). However, it does not provide significant improvements and relies on older neural network architectures because it was written in 2010 before the arrival of the Transformer architecture (Vaswani et al., 2017). Therefore, this approach was also disregarded.

[12] `https://huggingface.co/datasets/ctu-aic/csfever_v2`

3.8.2 NLI Filtering

Filtering using a well-performing, fine-tuned NLI model inspired by the idea by Ullrich et al. (2023) should yield a better dataset. Therefore, the following filtering schema was created, and its single steps are:

1. New labels and scores for the dataset are predicted using finetuned NLI model. This NLI model was trained by my supervisor Herbert Ullrich on the noisy dataset described in section 3.7.

2. Then the previously human-annotated gold data from Ullrich et al. (2023)[13] are used to optimize two threshold values. These threshold values tell us when to believe the prediction and when to use the label NOT ENOUGH INFO. One threshold is for the best F1 score and the second is for the precision. The optimization with its results is described in the subsection 3.8.4. In addition to the two thresholds, another one: 0.7, is tested for a more detailed view of the performance. All of the threshold values are computed after a calibration using temperature scaling, which makes the values more naturally interpretable. Temperature scaling itself is described in the subsection 3.8.3.

3. The data points below the thresholds are removed. Therefore, the dataset now consists only of verifiable data points. I made this decision because the dataset should be as clean of noise as possible, and the NOT ENOUGH INFO data points were not assigned 100 % correctly (see figure 3.3). The distribution of labels of filtered verifiable points is shown in section 3.10.

3.8.3 Temperature Scaling

The NLI model used for NLI filtering suffers, as nearly all modern neural networks from miscalibration. That means the NLI model is overconfident in its decisions. To treat this overconfidence, there exist calibration methods. According to the work of Guo et al. (2017), often the most effective, simplest and fastest is **temperature scaling**.

Temperature scaling does not affect the accuracy, however, it gives a better meaning for the found thresholds.[14] It is the simplest case of the Platt scaling. It computes the new prediction with the following formula:

$$q_i = \max_k \sigma_{SM}(z_i/T)^{(k)}$$

where q_i is a new confidence, T is the learned parameter called temperature that softens the softmax $\sigma_{SM}()$, and z_i is the logit vector for input i. The temperature is learned using negative log-likelihood and is tuned using the development split (The reason why only using development split is not so clear. Therefore, I conducted a small experiment where the temperature was tuned using also train split to see the difference. Results are in Appendix section B.1).

The implementation from Jan Drchal was used for learning the parameter T. The obtained parameter T for the model mentioned in step one in subsection 3.8.2 has a value of **1.9848**.

[13] 1 % of original CsFEVER data that were cross-examined by human annotators.
[14] Closer to the confidence of humans.

3.8.4 Threshold Optimization

The threshold value was optimized on the model calibrated using temperature scaling. The value of the threshold is a point of uncertainty when the model should classify the input as NOT ENOUGH INFO. Due to the lack of annotated data from the new dataset, previously human-annotated data made for CsFEVER by Ullrich (2021) were used.

The metrics chosen for the optimization are the F_1 **score** because it represents precision and recall in one number and **precision** because we want the optimization task to minimize the number of incorrectly predicted labels against gold labels.[15]

The threshold was optimized by minimizing the negative value of the metrics (among all three classes) using the function *fminbound()* from the Scipy library (Virtanen et al., 2020), and the computed optimal thresholds are:

- F_1: 0.888543
- **Precision**: 0.932376

The F_1 threshold was chosen to be first tried on the old annotated data. The computed confusion matrices are shown in Appendix B.2. Complete evaluation of new annotated data from section 3.9 is shown in section 3.11.

3.8.5 Other NLI Filtering Settings

Before submission, after discussing the results with my supervisor, another variant of NLI filtering was explored. In this variant, only the data points conflicting with the model's prediction were removed (keeping the data where low-confidence prediction matches the FEVER label) to lose fewer valid data points while still reducing the noise. This, however, did not lead to significant improvements in the experiments in chapter 4 over the simpler scheme from subsection 3.8.2 and is therefore omitted.

3.9 Annotations

To be able to measure how the filtering alters the dataset against gold labels[16], an annotation task was prepared using the *dev* and *train* splits of the noisy dataset described in section 3.7. The total number of claims is **133928** and it was decided to annotate approximately 1 % of randomly selected claims.[17] Moreover, another 200 claims were selected as the closest in terms of absolute value to the found threshold value because they are the most interesting for us to analyse as they are probably the hardest ones for the model's predictions.

The annotation was done using an annotation platform previously developed in (Ullrich, 2021).

[15] For precision the macro average was chosen as it is better for an unbalanced dataset. However, for F_1 score was experimentally chosen weighted average because its threshold is a little lower and therefore more data will survive the filtering.

[16] labels made by human annotators

[17] 1257 claims because the experiment was at first made on not unfolded dataset and because of that some annotated data are not useful.

3.10 Resulting Datasets

After the filtering, the distribution of labels in the datasets changed. Tables 3.6, 3.7, and 3.8 show the counts for F_1, precision and 0.7 thresholds, respectively. These counts are provided for data points with unfolded evidence sets. All three datasets are published in a Huggingface repository.[18]

	SUPPORTS	REFUTES	NEI	∑
train	35474	12325	35639	83438
dev	1131	981	3333	5445
test	1102	893	3333	5328
∑	37707	14199	42305	94211

Table 3.6: F1 threshold dataset - distribution of labels.

	SUPPORTS	REFUTES	NEI	∑
train	20187	5002	35639	60828
dev	593	362	3333	4288
test	602	301	3333	4236
∑	21382	5665	42305	69352

Table 3.7: Precision threshold dataset - distribution of labels.

	SUPPORTS	REFUTES	NEI	∑
train	54178	18790	35639	108607
dev	1824	1528	3333	6685
test	1822	1468	3333	6623
∑	57824	21786	42305	121915

Table 3.8: 0.7 threshold dataset - distribution of labels.

From tables 3.6, 3.7, and 3.8, we can see that from the 91623 verifiable data points, approximately **57 %** of them were preserved after the filtering with the F_1 threshold, **30 %** using the precision threshold, and **87 %** using the 0.7 threshold.

3.11 Evaluation

The performance of the new datasets against human-annotated data was measured using new annotated data from section 3.9.

Figure 3.2 shows a confusion matrix where labels from the original FEVER (the verifiable ones) are on the vertical axis, and annotated gold labels are on the horizontal axis.

[18] `https://huggingface.co/datasets/ctu-aic/csfever_v2`

This figure shows how the labels in the localized dataset should differ from the original labels.[19]

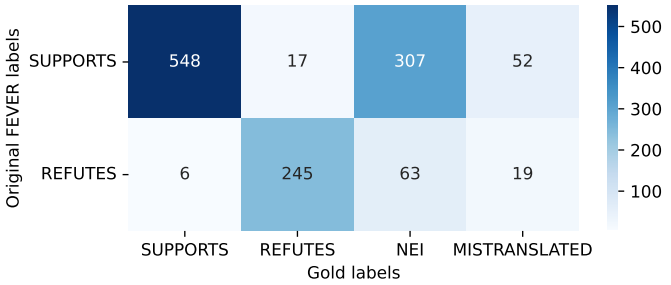


Figure 3.2: Confusion matrix of FEVER labels against annotated labels.

Figures 3.3 and 3.4 show confusion matrices of the counts after creating new datasets. On the vertical axes are the new labels of the datasets, and on the horizontal axes are the human-annotated gold labels. Figure 3.4 shows the labels when the NEI labels below the thresholds are thrown away, as described in section 3.8.2.

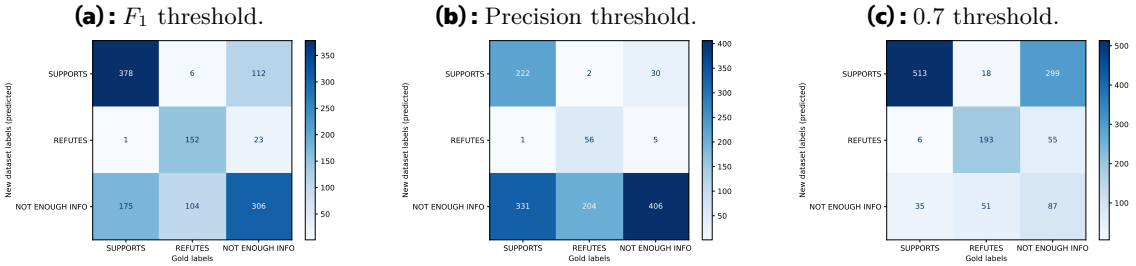


Figure 3.3: Confusion matrices of counts of labels of the new dataset against gold labels.

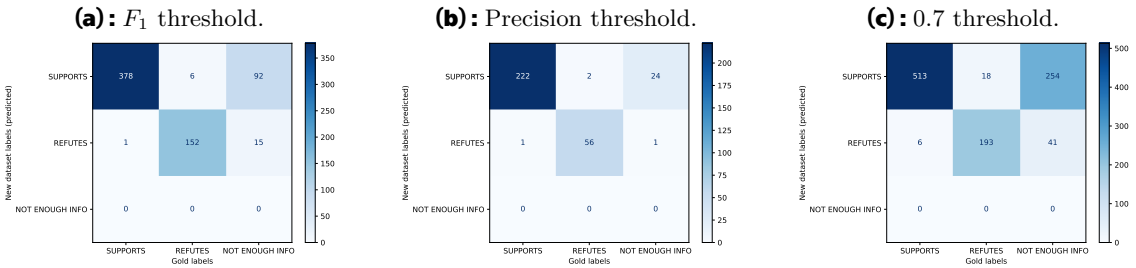


Figure 3.4: Confusion matrices of counts of labels of the new dataset without NEI labels below threshold against gold labels.

In the work of Ullrich et al. (2023) there was introduced a *Transduction precision* metric. This metric tells us the percentage of claims from all claims surviving the localization process that actually should survive.[20] The measured transduction precisions are approximately **82 %** for the F_1 threshold, **91 %** for the precision threshold, and **69 %** for the 0.7

[19]The considerable difference between the numbers of claims annotated as NEI that were originally labelled SUPPORTS and REFUTES occurred because the sample was chosen randomly, and the proportion of data points with SUPPORTS labels is higher in the original FEVER dataset.

[20]The claim that survives the localization process is a claim from the original FEVER, which was not filtered out. The *actually should survive* means that the labels of the new dataset are correct for these data points (According to the gold labels from the annotated data points).

threshold. When compared to transduction precision against the original FEVER labels which is **64 %** we can see that there is an improvement. However, it is needed to prove that this is really an improvement by training new models with these datasets in chapter 4 because this improvement also cost us some data points.

Moreover, also the annotated 200 claims nearest to the threshold were picked for empirical analysis, but no inherent patterns were found and there was nothing, in particular, to conclude from it.

Chapter 4
Model Training

This chapter describes the training and evaluation of document retrieval and NLI models, which is necessary to prove that the NLI filtering introduced in the previous chapter produces better datasets for Czech fact-checking.

4.1 Document Retrieval Models

In this section, selected solutions for document retrieval are discussed and evaluated. All implementations are in the enclosed repository.

4.1.1 Overview

It was necessary to choose a couple of the state-of-the-art document retrieval methods described in section 2.1. BM25 (4.1.1) was chosen as the baseline because, according to the work of Rýpar (2021), despite being older and even considered traditional, it provides a solid and efficient baseline for large-scale document retrieval.

The second one should use a dense approach method because they capture the semantic meaning. From the candidates listed in sections 2.1.2 and 2.1.3, currently the best for English information retrieval are SEAL and ColBERTv2. However, they were not ready at the time of writing this thesis for Czech document retrieval. Therefore, after discussing with my supervisor, we chose the hybrid approach, described in subsection 2.1.3, as the second document retrieval evaluated in this thesis.

BM25 - Anserini

For BM25, the **Anserini** implementation and its Python interface Pyserini developed by Lin et al. (2021), were chosen because it is a frequently used and reliable implementation. For the best results of the Anserini retrieval, it is necessary to set hyperparameters $k1$ and b. These hyperparameters were tuned using a grid search by Rýpar (2021), and the results were $k1 = 0.9$ and $b = 0.9$ for CsFEVER and $k1 = 0.6$ and $b = 0.5$ for ČTK. However, even though CsFEVER is much closer to the dataset created in this thesis, its hyperparameters yielded worse results and led to worse models than when using the best values for ČTK.[1] Therefore, the used parameters are $\mathbf{k1 = 0.6}$ and $\mathbf{b = 0.5}$. The examination of the influence of the choice of these hyperparameters and possibly running new grid search finetuning is left for future works.

[1] One possible explanation can be that these hyperparameters were finetuned on other metrics than MRR. However, MRR computation and especially dataset creation for NLI are dependent on the rank of the retrieved documents.

For the use of Anserini, it is also needed to pre-compute an index of all documents that can be retrieved.

Hybrid Retrieval

The Hybrid retrieval (described in subsection 2.1.3) was inspired by the work of Dědková (2021). For the first stage, called document preselection, was chosen the Anserini model. This model preselects 500 documents for the second stage, called document reranking. Due to the limited time, it was decided to use a pre-trained model from the Sentence Transformers library[2] for the second stage. Specifically, the chosen model is *cross-encoder/ms-marco-MiniLM-L-6-v2* because it has good results in metrics listed on Sentence Transformers evaluation page[3] while still having a good throughput of documents per second compared to the best model listed. For better results, it would be necessary to fine-tune the model on the datasets. The whole hybrid retriever, therefore, consists of **Anserini** and **ms-marco-MiniLM-L-6-v2** and is later in evaluation referred to as **Anserini+Cross-encoder**.

4.1.2 Evaluation

The document retrieval solutions Anserini and Anserini+Cross-encoder were evaluated and compared using MRR with $k \in \{1, 5, 10, 20\}$. MRR metric was described in section 2.4. The results are shown in table 4.1.

Dataset	Retriever	MRR@1	MRR@5	MRR@10	MRR@20
F_1	Anserini	32.53	43.53	44.87	45.28
	Anserini+Cross-encoder	**41.10**	**50.21**	**51.29**	**51.72**
Precision	Anserini	30.56	42.16	43.84	44.25
	Anserini+Cross-encoder	**41.31**	**50.63**	**51.80**	**52.25**
0.7	Anserini	32.92	43.30	44.58	45.02
	Anserini+Cross-encoder	**38.51**	**47.77**	**48.82**	**49.26**
Noisy	Anserini	29.59	38.5	39.65	40.02
	Anserini+Cross-encoder	**37.88**	**46.50**	**47.40**	**47.76**

Table 4.1: Document retrieval MRR evaluation (MRR@k in percentage, where k is the number of retrieved documents).

Table 4.1 shows that the hybrid retriever Anserini+Crossencoder outperformed the Anserini baseline retriever in all cases as was expected.

4.2 Natural Language Inference Models

This section describes the training of NLI models. First, a brief description of the selected approach is introduced. After that, the dataset preparation is described, then the training is briefly outlined, and the last subsection is dedicated to evaluation.

[2] https://www.sbert.net/examples/applications/retrieve_rerank/README.html
[3] https://www.sbert.net/docs/pretrained-models/ce-msmarco.html

4.2.1 Overview

The second part of the pipeline is an NLI model. The selected pre-trained model is **XLM-RoBERTa Large** finetuned on the SQuAD2 task by the Deepset company because it showed as the best performing in work by Ullrich et al. (2023). It was described in more detail in subsection 2.2.1.

This NLI model was then finetuned using four datasets: the noisy one without filtering and the three filtered described in subsection 3.8.2. The processing of the datasets for NLI finetuning is described in subsection 4.2.2. The model for the noisy dataset was previously fine-tuned by my supervisor in earlier stages of work. Therefore, it was used to save time and resources.

4.2.2 Dataset

For the finetuning of NLI models, it is necessary to preprocess the datasets described in section 3.10. Specifically, it is needed to acquire evidence for the NEI data points and process the evidence sets into strings. These strings are then with the strings of claims converted to a class prepared by the library for finetuning.[4]

The evidence for the non-verifiable data points was acquired similarly as in (Ullrich, 2021) by using the document retriever Anserini[5] to find the closest evidence from the Czech Wikipedia. (The Anserini was used instead of the better hybrid solution from section 4.1 because it was not ready when preparing the dataset.) Because the claim was not verifiable in the FEVER, which uses English Wikipedia, it also should not be verifiable in the Czech Wikipedia. Therefore, any chosen non-verifiable claim should not be verifiable by any evidence from Wikipedia.

The evidence was converted into strings simply by joining the single evidence texts with prepended titles in the order determined by the order in the list of evidence. This conversion is shown in table 4.2, where `'evidence'` is a field from data point that was created during the mapping step in section 3.5[6] and `'evidence_cs'` is then a Python dictionary where keys are the titles and values are Wikipedia articles found in the dump.

‘evidence’	[[‘Tygr indický’, ‘Bengal tiger’], [‘Tygr’, ‘Tiger’]]
‘evidence_cs’	{‘Tygr’: ‘Tygr (“Panthera tigris”) je velká kočkovitá šelma žijící v Asii. Ze …’, ‘Tygr indický’: ‘Tygr indický (“Panthera tigris tigris”), také zvaný tygr bengálský je nejpočetnější poddruh tygra. Vyskytuje se…’}
created evidence string	‘Tygr indický. Tygr indický (“Panthera tigris tigris”), také zvaný tygr bengálský je nejpočetnější poddruh tygra. Vyskytuje se... Tygr. Tygr (“Panthera tigris”) je velká kočkovitá šelma žijící v Asii. Ze ...’

Table 4.2: Evidence conversion example from `'evidence_cs'` from a data point to a new evidence string (evidence values were shortened using …) according to `'evidence'` order.

During the training of models, it turned out that the training results depended on the choice of Anserini hyperparameters and the evidence preprocessing, as seen in Appendix section B.3. These influences should be more thoroughly investigated in future works.

[4] In this thesis *InputExample* class from the Sentence Transformers library.
[5] The choice of hyperparameters is described in subsubsection 4.1.1.
[6] The first field is the Czech Wikipedia title, and the second is the English Wikipedia title.

4.2.3 Training

The models were finetuned with the help of the Sentence Transformers library[7] using a training script that was slightly modified from an example by Sentence Transformers library and also using a version previously adapted by my supervisor.

The training was done using different sets of hyperparameters. The tested hyperparameters were batch size (8 or 9) and different warmup ratios. However, after a few experiments, it was set to 0.4, as it led to the best-performing models in the majority of cases.

The models were trained for 20 epochs, and the best model was saved during the training according to the validation set evaluation. After three or four epochs, most models started overfitting (indicated by the decreasing validation accuracy). That means that the saved models were usually trained relatively soon. Graphs of the validation accuracies from the training of the best models are shown in Appendix section B.4. The whole training process was supervised using the Weights and Biases library[8] by Biewald (2020).

The best models are available for usage from the Huggingface repositories for each model: f1[9], precision[10], 0.7[11].

4.2.4 Evaluation

In table 4.3 are shown the resulting F_1 scores for the models trained on each dataset computed first on test splits of the dataset it was trained on. To make the table more readable, the full name of the pre-trained model **XLM-RoBERTa Large SQuAD2** is shortened to **XLM-RoBERTa**. For the comparison between models and also to investigate whether these models are overfitted, additional datasets were tried. These datasets are CsFEVER-NLI and CTKFactsNLI by Ullrich et al. (2023), ANLI by Nie et al. (2020) and SNLI by Bowman et al. (2015). The overfitting may have happened because the F_1 scores on test sets seem to be too high. I hypothesize that this could be caused by the filtering, where a similar model filters out the hard data points. Therefore, the dataset may not be challenging for the model.

	Test	CsFEVER-NLI	CTKFactsNLI	ANLI	SNLI
XLM-RoBERTa - Noisy	82.65	56.28	**69.70**	**32.46**	36.47
XLM-RoBERTa - F_1	92.49	**66.07**	58.08	26.26	**38.85**
XLM-RoBERTa - Precision	94.90	56.00	33.97	18.71	37.80
XLM-RoBERTa - 0.7	89.22	60.00	49.50	27.61	38.09

Table 4.3: NLI F_1 evaluation on test splits (macro F_1 score in percentage). The Test is the test split of dataset used to train each of the models (for each model is different).

Table 4.3 shows that the models trained on filtered datasets perform worse on datasets not close to their training set, such as CTKFactsNLI and better on datasets closer, such as CsFEVER-NLI[12]. The lower performance of all models on the SNLI dataset is probably caused by the much shorter evidence strings that people collected according to picture

[7] https://www.sbert.net/examples/training/nli/README.html
[8] https://wandb.ai/
[9] https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-csfever_v2-f1
[10] https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-csfever_v2-precision
[11] https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-csfever_v2-07
[12] Directly translated FEVER dataset.

descriptions. The models have lower performance on ANLI, probably because ANLI was made to be challenging for state-of-the-art models.

To have a comparison with results from (Ullrich et al., 2023), some results from that paper and new results on ANLI and SNLI datasets that Herbert Ullrich provided are shown in table 4.4.

	CsFEVER-NLI	ANLI	SNLI
M-BERT base - CsFEVER	_	28.3	36.2
XLM-RoBERTa @ SQUAD2 - CsFEVER	_	32.4	39.9
XLM-RoBERTa @ SQUAD2 - CsFEVER-NLI	72.2	34.1	56.9
XLM-RoBERTa @ XNLI - CsFEVER-NLI	73.7	32.6	57.3

Table 4.4: NLI F_1 evaluation of models from (Ullrich et al., 2023) on test splits. Results were provided by Herbert Ullrich. (macro F_1 score in percentage) The missing results "_" for CsFEVER-NLI could not be computed because of massive leakage in the datasets.

To better illustrate the causes of lower F_1 scores in table 4.3, confusion matrices for the CTKFactsNLI dataset[13] are shown in figure 4.1. This figure shows that all new models set NEI labels more often than the previously trained model on the noisy dataset, which also assigns other labels. This new behaviour is preferred because it is better for the model in the fact-checking pipeline or application to mismark the claim as NEI than choose the wrong label between the verifiable ones.

In conclusion, the filtering may allow us to finetune slightly better models on specific fact-checking tasks in the Czech language and low-resource languages. However, the reason why it has a lower F_1 score on some datasets should be further investigated.

[13]This dataset was chosen as the most descriptive one because of the lower performance of the new datasets on it.

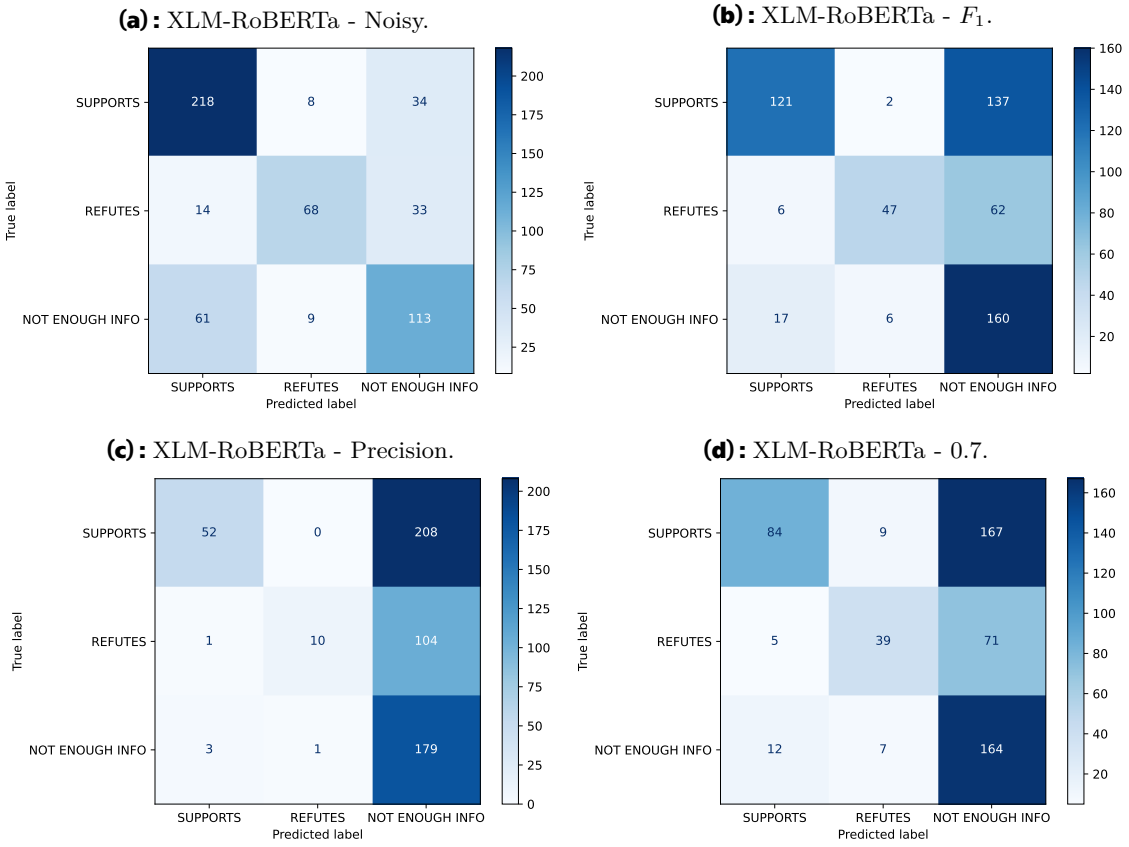


Figure 4.1: Confusion matrices of models predictions on the test sets of CTKFactsNLI.

Chapter 5
Fact-checking Pipeline

This chapter describes the features of the initial version of the fact-checking pipeline. First, the pipeline is described then some additional features are described and at the end of the chapter, the whole pipeline is evaluated.

5.1 Overview

The fact-checking pipeline structure is taken over from the work by Ullrich (2021). A simple diagram of the pipeline is shown in figure 5.1. This pipeline consists of two main stages. The first stage is document retrieval and the second stage is NLI. The corresponding models were prepared in chapter 4.

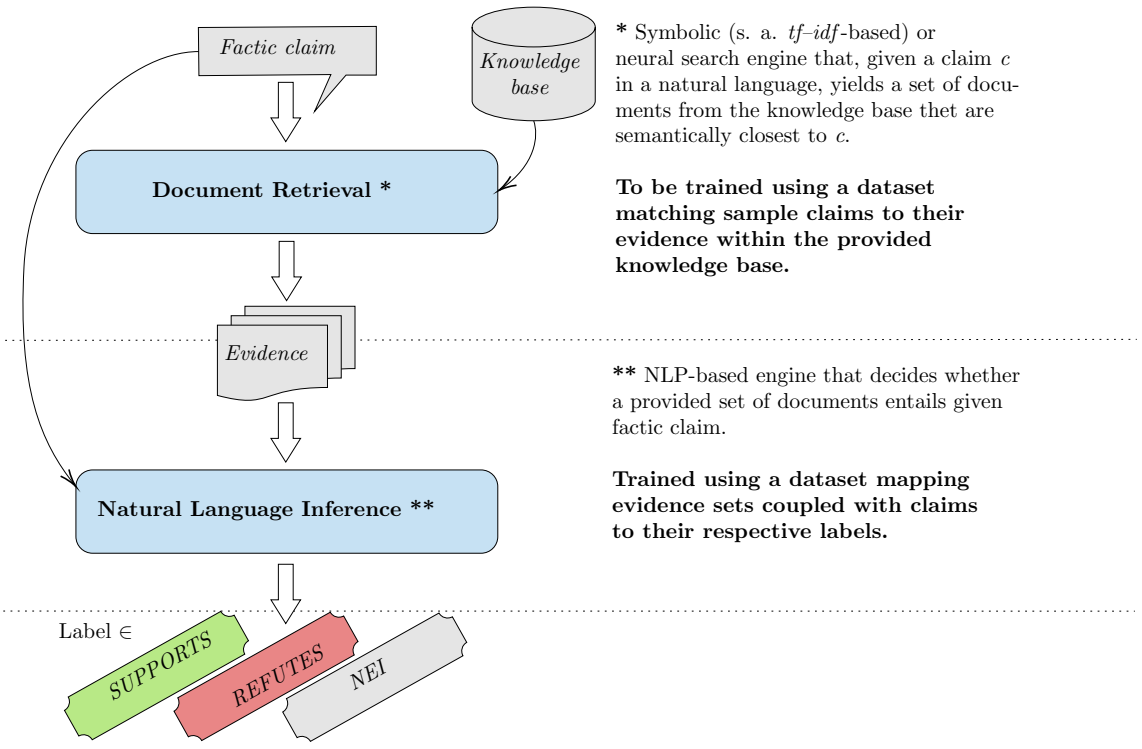


Figure 5.1: Fact-checking pipeline, reprinted from Ullrich (2021).

In this pipeline, the claim first enters the document retrieval phase where a document retriever retrieves evidence from a knowledge base. In this thesis, the knowledge base is a Wikipedia dump described in chapter 3. This retrieved evidence is then passed to the NLI phase where a label is assigned to the given claim with respect to it.

In addition to this basic pipeline, some new features are introduced in the next two sections. The features are not a necessary part of the pipeline itself. However, they can be integrated into it to provide the user with a better understanding of the decision.

5.2 Explainability

From the decisions made by an NLI model, it is not evident why the model decided this way or what part of the input influenced it the most. The influence of input parts can be investigated using explainability methods. The explainability can be helpful for fact-checkers because they can immediately see the essential parts of the evidence.

The explainability of NLI models for Czech fact-checking was explored by Kopecká (2022). In this thesis two main explainability methods SHAP[1] by Lundberg et al. (2017) and LIME[2] by Ribeiro et al. (2016), were pursued.

Kopecká (2022) tested that SHAP produced better results than LIME. Therefore, SHAP implementation from that thesis was added as an option to the pipeline. SHAP determines the contribution of each part of input (the granularity) to the final score. The maximum number of evaluations then influences the resolution of the explanation and influences the computational time. The best SHAP setting from (Kopecká, 2022) was sampling approximation, max 5000 evaluations, and word granularity. However, 5000 evaluations take too much time for the application. Therefore, the number of evaluations must be lowered in the showcase application. An example of the explainability in NLI task is shown in figure 5.2.

5.3 Temperature Scaling

Temperature scaling was previously described in subsection 3.8.3. Therefore, this section only introduces the new values of the parameter T[3] for the new datasets. These values are shown in table 5.1[4]. These values can then be used when the confidences from the NLI phase are presented to the user.

NLI model	Temperature T
XLM-RoBERTa - Noisy	1.9848
XLM-RoBERTa - F_1	2.0698
XLM-RoBERTa - Precision	2.1136
XLM-RoBERTa - 0.7	2.0356

Table 5.1: Learned temperatures for the NLI models.

5.4 Evaluation

The pipeline was evaluated by the script by Jan Drchal to directly compare with the work by Ullrich et al. (2023). The experiments were carried out in the same setting (described

[1] `https://shap.readthedocs.io/`
[2] `https://lime-ml.readthedocs.io/`
[3] temperature
[4] The names of NLI models use the same convention as in chapter 4.

Fakulta elektrotechnická ČVUT (FEL ČVUT) je fakulta ČVUT s cca 3 100 studenty , 730 zaměstnanci a ročním rozpočtem přesahujícím 800 milionů korun . Poslání fakulty . Elektrotechnická fakulta ČVUT vychovává odborníky v oblasti elektrotechniky , energetiky , softwarového inženýrství , sdělovací techniky , robotiky a kybernetiky , automatizace , informatiky a výpočetní techniky . Je také centrem pro vědeckou a výchovnou činnost v uvedených oblastech . Studijní programy . Fakulta elektrotechnická uskutečňuje výuku ve studijních programech . V prvním ročníku si již vybírají studenti obor : Věda a výzkum . Fakulta je jedním z největších výzkumných pracovišť v ČR , (pátým dle aktuálního hodnocení Rady vlády pro výzkum a vývoj) . Počítačové studovny . Počítačové studovny s volným přístupem v Dejvicích provozuje oddělení výpočetní techniky Střediska vědeckotechnických informací (SVTI) . Na Karlově náměstí se také nachází několik místností s počítači se systémem Windows nebo Solaris . Samozřejmostí je možnost Wi-Fi připojení Eduroam . Katedry . Výuka i výzkum jsou na fakultě organizovány katedrami , tj . specializovanými pracovišti . Katedry fakulty mají přidělený alfanumerický kód K131xx , jednoznačný v rámci celé univerzity . Symbol `` xx '' představuje dvojciferné číslo , pod kterým katedra vystupuje v rámci fakulty . Toto číslo je součástí kódů vyučovaných předmětů , čímž přispívá k jednoznačnému určení předmětu dle kódu . K 1. srpnu 2007 působilo na fakultě 17 kateder , jedno centrum a jedno středisko . V akademickém roce 2006/2007 zanikla `` Katedra tělesné výchovy '' . Výuku tělesné výchovy zajišťuje od akademického roku 2007/2008 `` Ústav tělesné výchovy a sportu ČVUT '' (ÚTVS ČVUT) . Studentská konference POSTER . FEL ČVUT každoročně v květnu pořádá studentskou konferenci POSTER , na které jsou prezentovány výsledky práce studentů a doktorandů . Tato konference je vynikající příležitostí k setkání s aktivními a profesně zdatnými studenty . Zhruba čtvrtina příspěvků je zahraničních . Spolek absolventů . Absolventi FEL se sdružují ve spolku ELEKTRA . Spolek pořádá každoroční srazy absolventů , koncerty a další akce .

Figure 5.2: SHAP explanation example for the label with the highest confidence for claim *FEL ČVUT je fakulta ČVUT.* (In English: FEE CTU is a faculty of CTU.). The more a word is highlighted in red, the more it sways the score for this decision and the more it is blue, the more it influences against it.

below), to give comparable results.

The evaluation consists of the following phases, and the description was adapted from Ullrich et al. (2023):

1. Firstly, k documents are retrieved $D = \{d_1, \ldots, d_k\}$ by a given retrieval model (in the table denoted as `@k`). These documents in set D are ordered by decreasing relevancy.

2. Then, 2 scenarios are considered:

   - **Score Evidence (SE)** means that the retrieved set D must fully cover the set of gold evidence G given by the test dataset ($G \subseteq D$). The data point is treated as wrong if this condition is not met. In the other case, the NLI prediction is done as described later. The data points with NEI labels are automatically treated as wrong because no gold evidence is provided.
   - **No Score Evidence (NSE)** means that no such condition as in *SE* is set, and all data points are used for NLI prediction and evaluation.

3. Then because of the limiting size of the input of NLI models[5], the documents cannot be concatenated. They are split into l consecutive splits $S = \{s_1, \ldots, s_l\}$ where $l \leq k$. Each split $s_i, \forall i = 1, \ldots, l$ is formed from one document or concatenation of

[5]Used NLI models are XLM-RoBERTa which has a limited maximal input size of 512 tokens.

documents $s_i = \{d_s, \ldots, d_e\}$, where $1 \leq s \leq e \leq k$. A new split s_{i+1} is created in three cases. First, if a new document d_j that would be added into split s_i would be over the size limit for the NLI model. The second case is when a single document d_j that would create a new split exceeds the input size. In this case, the document is truncated to the maximum size and is represented as a single split. Due to the length of documents in this thesis, this happened most of the time. The last case is when the split reaches the maximum number of evidence documents in it – k_s, which was set to $k_s = 2$. This limit was taken over from Ullrich et al. (2023), where it serves to have a similar average length for different datasets.

4. In the next phase, the split's documents $d_s, \ldots, d_e$ are concatenated, and all concatenated splits are passed to the NLI model along with the claim provided by the dataset. Therefore, the NLI model returns confidences $y_1 \ldots y_l$ where $y_i = \{y_i^{\text{SUPPORTS}}, y_i^{\text{REFUTES}}, y_i^{\text{NEI}}\}$. The resulting confidences are then computed as a weighted average:

$$y^c = \frac{1}{l} \sum_{i=1}^{l} \lambda^{i-1} y_i^c, c \in \{\text{SUPPORTS}, \text{REFUTES}, \text{NEI}\}.$$

where λ is a hyperparameter that was set to $\frac{1}{2}$ as in Ullrich et al. (2023) that weights the average to give higher importance to the higher-ranked retrieved documents.

5. Then the argmax is computed from y^c, and the label is assigned[6]. With these assignments are then computed F_1 macro scores for each case which are shown in table 5.2.

Dataset/ NLI model	Retrieval	@1		@5		@10		@20	
		NSE	SE	NSE	SE	NSE	SE	NSE	SE
F_1	Anserini	61.19	12.95	61.52	16.77	52.92	13.49	45.01	9.18
	Hybrid	**66.40**	*17.10*	*65.95*	**20.77**	*62.83*	*19.38*	*52.11*	*13.66*
Precision	Anserini	61.85	9.01	62.06	12.38	56.71	12.33	43.89	5.48
	Hybrid	**69.91**	*13.43*	*69.07*	*16.75*	68.31	**17.68**	51.75	*9.70*
0.7	Anserini	54.01	13.36	53.83	15.81	35.35	6.05	33.49	5.58
	Hybrid	**58.45**	*16.18*	*57.96*	**18.98**	*45.47*	*11.32*	*38.76*	*8.67*

Table 5.2: Full pipeline F_1 evaluation (macro F_1 score in percentage). The bold numbers represent the best score for NSE and SE within one dataset (and therefore NLI model) and the numbers in italics represent the better scores for each column within one dataset. **Hybrid** is used instead Anserini+Cross-encoder.

Table 5.2 shows that all models best perform for SE evaluation around 5-10 retrieved documents. Therefore, a number from this interval will be the default option in the showcase application. For NSE evaluation, the best score is always for one retrieved document. This behaviour was expected because the first retrieved document should be the most helpful. The others, especially for more retrieved documents (10, 20), could influence the result badly (probably to an NEI decision). The results are partially loosely

[6]In the case of SE are here chosen wrong labels for NLI data points and for data points that did not meet the condition.

comparable to results on different datasets and splits in (Ullrich et al., 2023). This table again shows that the Hybrid retrieval solution (Anserini+Cross-encoder) outperforms the Anserini as expected. Thus it could serve as a new baseline for all retrieval tasks in the Czech language.

Chapter 6
Showcase Application

A showcase application is valuable and necessary to present the fact-checking pipeline to other people without technical knowledge, such as journalists and fact-checkers. It should provide a simple Graphical User Interface (GUI) to examine the pipelines and their properties.

As mentioned in the work of Ullrich (2021), Jan Drchal created a showcase application in the Dash framework called *Fact Search*. However, it was built in an older version of the Dash platform[1]. That caused conflicts in dependencies when newer versions of libraries and also newer Python versions were used. It took much work to run it and adapt it to use the newer versions of the used libraries. Therefore, I explored other platforms for Python applications and other possibilities. These possible approaches were examined for their advantages and disadvantages and will be described in section 6.1.

6.1 Possible Approaches

As mentioned earlier, finding another solution was necessary. The most complex solution would be creating a new application from scratch (possibly using the React[2] and Flask[3] libraries). However, it could be better to use a prepared Python framework because this thesis is not aimed at software engineering, and it would be hard to maintain the application in time. These frameworks are being continuously developed, with new features coming nearly every month and are much simpler and easier to use. Therefore, they are a good choice for simple showcase applications, as is the one developed in this thesis. After some research, I narrowed the possibilities to the *Dash*, *Gradio*[4] and *Streamlit*[5] platforms.

6.1.1 Dash

The Dash Open Source framework is one of the well-known UI Python libraries introduced by the Plotly company. It is a robust framework that allows users to build GUI web applications entirely in Python. It also provides significant customization, making it possible to develop customized components. However, developing such components is not very straightforward because JavaScript and React are needed. Another disadvantage is that it is an extensive framework that is not easy to use. As said earlier, Jan Drchal already tried Dash for this kind of application. Thus I preferred trying other alternatives

[1] `https://plotly.com/dash/`
[2] `https://react.dev/`
[3] `https://flask.palletsprojects.com/`
[4] `https://gradio.app/`
[5] `https://streamlit.io/`

to directly compare the code complexity and the options that these alternatives provide for building the wanted interface.

6.1.2 Gradio

Gradio is a relatively new framework by Abid et al. (2019). It aims to build simple machine learning apps as fast as possible and works excellently directly in Jupyter notebooks. One of the most significant advantages is its direct connection to the Huggingface projects[6], including Huggingface Spaces, where these applications can be hosted. All of this makes Gradio perfectly suitable for simple applications. Therefore, I first decided to choose this framework for this thesis (screenshots of the app are in Appendix C, and the code is in the enclosed repository). However, I discovered many limitations during the work due to its simplicity and was forced to try alternatives. One of the limitations was its inability to change the interface after creating it. Another disadvantage is that it is problematic to develop new components. These disadvantages caused problems with outputting more than one result from the pipeline. Apart from these problems, this framework is a good choice for developing simple small applications, especially in NLP or computer vision.

6.1.3 Streamlit

Streamlit is another known framework for building Python applications. Its disadvantage is a difference in the deployment workflow. The app should be developed in a standard Python file and reloaded whenever something changes. This reloading would cause problems loading big models and large amounts of data, and therefore Streamlit provides some caching features. Its code is more straightforward than the code of the Dash framework but provides more options than the Gradio framework (including solutions to all Gradio's disadvantages). Therefore, despite the disadvantage, I decided to use this framework for the showcase application in this thesis.

6.2 Showcase Application

The prototype showcase application made with the Streamlit framework is shown in figure 6.1. The code is available in the enclosed repository, additional screenshots of the functions are shown in Appendix C, and a static demo is available on Streamlit Cloud[7]. The features of this application are described in the next section.

6.2.1 Features

This subsection describes the features of the developed showcase application. All of them are displayed in figure 6.1.

A fact-checking application must have a field for entering the claim to be fact-checked, and then upon user input, it should run the pipeline described in chapter 5, and then display its results.

Specifically, in the prototype developed as part of this thesis called *Fact-checking platform Streamlit GUI*, a user enters the claim into the field **claim**. Then the user can choose how many results should be retrieved by the field **Number of results** and choose

[6] `https://huggingface.co/`
[7] `https://fact-check-platform.streamlit.app/`

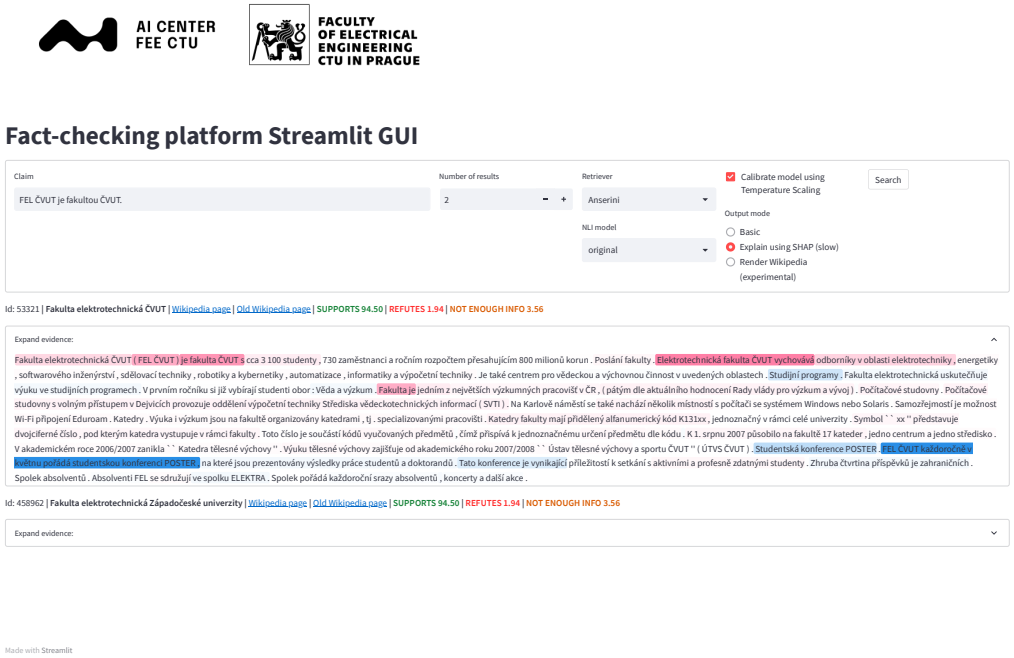


Figure 6.1: Streamlit GUI - example of its functions.

whether they want to use the calibration by checking **Calibrate model using Temperature Scaling**. Subsequently, they can choose what **Output mode** should be used. The **Basic** mode shows only the retrieved text, the **Explain using SHAP (slow)** mode shows the SHAP explanation, and the **Render Wikipedia** mode embeds the Wikipedia page used for the decision. After entering the claim and choosing desired options, the user should click the button **Search**, and then the results are displayed below. Each result comprises seven information fields and an expander showing the retrieved evidence. The information fields are:

- **Id**, which is used for debugging purposes,
- the title of the Wikipedia page,
- link to the current Wikipedia page – **Wikipedia page**, link to the Wikipedia page which was originally the source of the retrieved evidence (specific revision of the Wikipedia page) – **Old Wikipedia page**,
- predicted percentages for each of the possible labels (**SUPPORTS**, **REFUTES**, **NOT ENOUGH INFO**).

The expander then shows the output in the desired output mode.

Chapter 7
Conclusion

In this bachelor thesis, I explored state-of-the-art methods for document retrieval and NLI in Czech language (chapter 2). Then I acquired and preprocessed a new Czech Wikipedia dataset (chapter 3). Next, filtering was applied to this dataset resulting in three new datasets. Two of them were filtered using threshold maximising F_1 score and precision. The third one was filtered with a threshold of 0.7 (chapter 3). I trained and evaluated new models for document retrieval and NLI with these new datasets (chapter 4). Then I combined these new models, the explainability method SHAP and temperature scaling, into an initial version of the fact-checking pipeline (chapter 5). As the last step, I created a prototype showcase application for the pipeline (chapter 6). This thesis is based on recent works at Artificial Intelligence Center FEE CTU by Ullrich (2021), Rýpar (2021), Dědková (2021), Kopecká (2022) and Ullrich et al. (2023). This thesis adds new insight into the works, combines selected results and addresses the NLI filtering proposal.

The filtering for all three thresholds increased the transduction precision in datasets. However, as chapter 4 shows, the filtering did not increase the model's performance on all datasets. Overall, the dataset F_1 performed the best from the newly filtered datasets and can be helpful in specific fact-checking tasks. It also prioritised the wanted NEI label in uncertain cases, thus allowing us to train NLI models with better behaviour. However, as a whole, the NLI filtering did not show as a promising approach for future works. The evaluation of document retrieval models showed that the hybrid solution, Anserini combined with the Cross-encoder model, outperforms Anserini in all cases. Therefore, hybrid retrieval could be a more robust baseline for Czech document retrieval than simple Anserini (BM25). In the pipeline, SHAP and temperature scaling showed their ability to partially solve the problem of the explainability and calibration of NLI models. The Streamlit library has then proven to provide an ideal mixture of simplicity and available options for building the prototype showcase application.

Automated Czech fact-checking naturally has its limitations. I tried to deal with the noise within the localised dataset using the NLI filtering, but it filtered out the noise only partially and led to a loss in the number of available data points. The training then showed that the new models tend to overfit on these new datasets. I hypothesise that the overfitting occurred because the model used for filtering filtered out the data points that were difficult for it to evaluate. Therefore, the resulting dataset was not so challenging for the training and led to overly promising results on test splits.

In future works, the applications of the introduced pipeline and approach to low-resource languages should be investigated to prove that they can also be applied to other languages. Also, the performance of the recently arriving LPLMs on fact-checking tasks should be analysed. Additionally, the effect of hyperparameter selection in BM25 retrieval and the influence of dataset preprocessing on the training of NLI models should be thoroughly examined because, in this thesis, they significantly influenced the results. Finally, a new

method, zero-shot fact verification by Pan et al. (2021), should be explored because it offers a solution to the limitations of the dataset localisation approach described in this thesis to obtain the Czech dataset for fact-checking. This new approach artificially generates claims and labels directly from the desired Wikipedia dump, preventing the localisation process from introducing noise to the dataset.

Bibliography

ABID, Abubakar; ABDALLA, Ali; ABID, Ali; KHAN, Dawood; ALFOZAN, Abdulrahman; ZOU, James, 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569*.

ARKHIPOV, Mikhail; TROFIMOVA, Maria; KURATOV, Yuri; SOROKIN, Alexey, 2019. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In: *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*. Florence, Italy: Association for Computational Linguistics, pp. 89–93. Available from DOI: 10.18653/v1/W19-3712.

ATTARDI, Giusepppe, 2015. *WikiExtractor* [https://github.com/attardi/wikiextractor]. GitHub.

BEVILACQUA, Michele; OTTAVIANO, Giuseppe; LEWIS, Patrick; YIH, Wen-tau; RIEDEL, Sebastian; PETRONI, Fabio, 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. In: *arXiv pre-print 2204.10628*. Available also from: https://arxiv.org/abs/2204.10628.

BIEWALD, Lukas, 2020. *Experiment Tracking with Weights and Biases*. Available also from: https://www.wandb.com/. Software available from wandb.com.

BOWMAN, Samuel R.; ANGELI, Gabor; POTTS, Christopher; MANNING, Christopher D., 2015. A large annotated corpus for learning natural language inference. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 632–642. Available from DOI: 10.18653/v1/D15-1075.

BROWN, Tom; MANN, Benjamin; RYDER, Nick; SUBBIAH, Melanie; KAPLAN, Jared D; DHARIWAL, Prafulla; NEELAKANTAN, Arvind; SHYAM, Pranav; SASTRY, Girish; ASKELL, Amanda; AGARWAL, Sandhini; HERBERT-VOSS, Ariel; KRUEGER, Gretchen; HENIGHAN, Tom; CHILD, Rewon; RAMESH, Aditya; ZIEGLER, Daniel; WU, Jeffrey; WINTER, Clemens; HESSE, Chris; CHEN, Mark; SIGLER, Eric; LITWIN, Mateusz; GRAY, Scott; CHESS, Benjamin; CLARK, Jack; BERNER, Christopher; MCCANDLISH, Sam; RADFORD, Alec; SUTSKEVER, Ilya; AMODEI, Dario, 2020. Language Models are Few-Shot Learners. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.F.; LIN, H. (eds.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Vol. 33, pp. 1877–1901. Available also from: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

ČESKO V DATECH, 2022. *Dezinformace - České dezinformační weby loni publikovaly téměř 200 tisíc článků. Češi si však v jejich rozpoznávání stále dostatečně nevěří* [online] [visited on 2023-03-22]. Available from: `https://www.ceskovdatech.cz/clanek/176-dezinformace/`.

CHEN, Danqi; FISCH, Adam; WESTON, Jason; BORDES, Antoine, 2017. Reading Wikipedia to Answer Open-Domain Questions. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1870–1879. Available from DOI: `10.18653/v1/P17-1171`.

CHOWDHERY, Aakanksha; NARANG, Sharan; DEVLIN, Jacob; BOSMA, Maarten; MISHRA, Gaurav; ROBERTS, Adam; BARHAM, Paul; CHUNG, Hyung Won; SUTTON, Charles; GEHRMANN, Sebastian; SCHUH, Parker; SHI, Kensen; TSVYASHCHENKO, Sasha; MAYNEZ, Joshua; RAO, Abhishek; BARNES, Parker; TAY, Yi; SHAZEER, Noam; PRABHAKARAN, Vinodkumar; REIF, Emily; DU, Nan; HUTCHINSON, Ben; POPE, Reiner; BRADBURY, James; AUSTIN, Jacob; ISARD, Michael; GUR-ARI, Guy; YIN, Pengcheng; DUKE, Toju; LEVSKAYA, Anselm; GHEMAWAT, Sanjay; DEV, Sunipa; MICHALEWSKI, Henryk; GARCIA, Xavier; MISRA, Vedant; ROBINSON, Kevin; FEDUS, Liam; ZHOU, Denny; IPPOLITO, Daphne; LUAN, David; LIM, Hyeontaek; ZOPH, Barret; SPIRIDONOV, Alexander; SEPASSI, Ryan; DOHAN, David; AGRAWAL, Shivani; OMERNICK, Mark; DAI, Andrew M.; PILLAI, Thanumalayan Sankaranarayana; PELLAT, Marie; LEWKOWYCZ, Aitor; MOREIRA, Erica; CHILD, Rewon; POLOZOV, Oleksandr; LEE, Katherine; ZHOU, Zongwei; WANG, Xuezhi; SAETA, Brennan; DIAZ, Mark; FIRAT, Orhan; CATASTA, Michele; WEI, Jason; MEIER-HELLSTERN, Kathy; ECK, Douglas; DEAN, Jeff; PETROV, Slav; FIEDEL, Noah, 2022. *PaLM: Scaling Language Modeling with Pathways*. Available from arXiv: `2204.02311 [cs.CL]`.

CONNEAU, Alexis; KHANDELWAL, Kartikay; GOYAL, Naman; CHAUDHARY, Vishrav; WENZEK, Guillaume; GUZMÁN, Francisco; GRAVE, Edouard; OTT, Myle; ZETTLEMOYER, Luke; STOYANOV, Veselin, 2020. Unsupervised Cross-lingual Representation Learning at Scale. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. Available from DOI: `10.18653/v1/2020.acl-main.747`.

DĚDKOVÁ, Barbora, 2021. *Multi-stage Methods for Document Retrieval in the Czech Language*. Available also from: `http://hdl.handle.net/10467/97064`. MA thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina, 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. Available from DOI: `10.18653/v1/N19-1423`.

GUO, Chuan; PLEISS, Geoff; SUN, Yu; WEINBERGER, Kilian Q., 2017. On Calibration of Modern Neural Networks. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. Sydney, NSW, Australia: JMLR.org, pp. 1321–1330. ICML'17.

JEATRAKUL, Piyasak; WONG, Kok Wai; FUNG, Chun Che, 2010. Data Cleaning for Classification Using Misclassification Analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*. Vol. 14, no. 3, pp. 297–302. Available from DOI: 10.20965/jaciii.2010.p0297.

KHATTAB, Omar; ZAHARIA, Matei, 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Virtual Event, China: Association for Computing Machinery, pp. 39–48. SIGIR '20. ISBN 9781450380164. Available from DOI: 10.1145/3397271.3401075.

KOPECKÁ, Eliška, 2022. *Explaining NLP Model Predictions for Fact-Checking Pipeline*. Available also from: http://hdl.handle.net/10467/101307. MA thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

LEE, Nayeon; LI, Belinda Z.; WANG, Sinong; YIH, Wen-tau; MA, Hao; KHABSA, Madian, 2020. Language Models as Fact Checkers? In: *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Online: Association for Computational Linguistics, pp. 36–41. Available from DOI: 10.18653/v1/2020.fever-1.5.

LIN, Jimmy; MA, Xueguang; LIN, Sheng-Chieh; YANG, Jheng-Hong; PRADEEP, Ronak; NOGUEIRA, Rodrigo, 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In: *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pp. 2356–2362.

LIU, Yinhan; OTT, Myle; GOYAL, Naman; DU, Jingfei; JOSHI, Mandar; CHEN, Danqi; LEVY, Omer; LEWIS, Mike; ZETTLEMOYER, Luke; STOYANOV, Veselin, 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. Available from arXiv: 1907.11692 [cs.CL].

LUAN, Yi; EISENSTEIN, Jacob; TOUTANOVA, Kristina; COLLINS, Michael, 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*. Vol. 9, pp. 329–345. Available from DOI: 10.1162/tacl_a_00369.

LUNDBERG, Scott M; LEE, Su-In, 2017. A Unified Approach to Interpreting Model Predictions. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (eds.). *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 4765–4774. Available also from: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

MACCARTNEY, Bill; MANNING, Christopher D., 2008. Modeling Semantic Containment and Exclusion in Natural Language Inference. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 521–528. Available also from: https://aclanthology.org/C08-1066.

MANNING, Christopher D., 2022. Human Language Understanding & Reasoning. Available also from: https://www.amacad.org/publication/human-language-understanding-reasoning.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; HINRICH, Schütze, 2008. *Introduction to Information Retrieval*. Cambridge University Press. ISBN 0521865719.

MEHDI, Yusuf, 2023a. *Confirmed: the new Bing runs on OpenAI's GPT-4* [online] [visited on 2023-03-15]. Available from: https://blogs.bing.com/search/march_2023/Confirmed-the-new-Bing-runs-on-OpenAI%E2%80%99s-GPT-4.

MEHDI, Yusuf, 2023b. *Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web* [online] [visited on 2023-02-26]. Available from: https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/.

NIE, Yixin; WILLIAMS, Adina; DINAN, Emily; BANSAL, Mohit; WESTON, Jason; KIELA, Douwe, 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

OPENAI, 2022. *Introducing ChatGPT* [online] [visited on 2023-03-30]. Available from: https://openai.com/blog/chatgpt.

OPENAI, 2023. *GPT-4 Technical Report*. Available from arXiv: 2303.08774 [cs.CL].

PAN, Liangming; CHEN, Wenhu; XIONG, Wenhan; KAN, Min-Yen; WANG, William Yang, 2021. Zero-shot Fact Verification by Claim Generation. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, pp. 476–483. Available from DOI: 10.18653/v1/2021.acl-short.61.

PAPINENI, Kishore; ROUKOS, Salim; WARD, Todd; ZHU, Wei-Jing, 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. Available from DOI: 10.3115/1073083.1073135.

PLEISS, Geoff; ZHANG, Tianyi; ELENBERG, Ethan; WEINBERGER, Kilian Q., 2020. Identifying Mislabeled Data Using the Area under the Margin Ranking. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates Inc. NIPS'20. ISBN 9781713829546.

QU, Yingqi; DING, Yuchen; LIU, Jing; LIU, Kai; REN, Ruiyang; ZHAO, Wayne Xin; DONG, Daxiang; WU, Hua; WANG, Haifeng, 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 5835–5847. Available from DOI: 10.18653/v1/2021.naacl-main.466.

RAJPURKAR, Pranav; JIA, Robin; LIANG, Percy, 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 784–789. Available from DOI: 10.18653/v1/P18-2124.

RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos, 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA: Association for Computing Machinery, pp. 1135–1144. KDD '16. ISBN 9781450342322. Available from DOI: 10.1145/2939672.2939778.

ROBERTSON, Stephen; ZARAGOZA, Hugo, 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* Vol. 3, no. 4, pp. 333–389. ISSN 1554-0669. Available from DOI: 10.1561/1500000019.

RÝPAR, Martin, 2021. *Methods of Document Retrieval for Fact Checking*. Available also from: http://hdl.handle.net/10467/95315. MA thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

SANTHANAM, Keshav; KHATTAB, Omar; SAAD-FALCON, Jon; POTTS, Christopher; ZAHARIA, Matei, 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 3715–3734. Available from DOI: 10.18653/v1/2022.naacl-main.272.

SIDO, Jakub; PRAŽÁK, Ondřej; PŘIBÁŇ, Pavel; PAŠEK, Jan; SEJÁK, Michal; KONOPÍK, Miloslav, 2021. Czert – Czech BERT-like Model for Language Representation. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. Held Online: INCOMA Ltd., pp. 1326–1338. Available also from: https://aclanthology.org/2021.ranlp-1.149.

SPEER, Robyn, 2019. *ftfy* [Zenodo]. Available from DOI: 10.5281/zenodo.2591652. Version 5.5.

STRAKA, Milan; NÁ PLAVA, Jakub; STRAKOVÁ, Jana; SAMUEL, David, 2021. RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. In: *Text, Speech, and Dialogue*. Springer International Publishing, pp. 197–209. Available from DOI: 10.1007/978-3-030-83527-9_17.

TALUKDAR, Arka; DAGAR, Monika; GUPTA, Prachi; MENON, Varun, 2021. Training Dynamic based data filtering may not work for NLP datasets. In: *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 296–302. Available from DOI: 10.18653/v1/2021.blackboxnlp-1.22.

TETLOCK, Philip, 2019. *Superforecasting: The Art and Science of Prediction*. Random House Bussiness. ISBN 9781847947154.

THOPPILAN, Romal; FREITAS, Daniel De; HALL, Jamie; SHAZEER, Noam; KULSHRESHTHA, Apoorv; CHENG, Heng-Tze; JIN, Alicia; BOS, Taylor; BAKER, Leslie; DU, Yu; LI, YaGuang; LEE, Hongrae; ZHENG, Huaixiu Steven; GHAFOURI, Amin; MENEGALI, Marcelo; HUANG, Yanping; KRIKUN, Maxim; LEPIKHIN, Dmitry; QIN, James; CHEN, Dehao; XU, Yuanzhong; CHEN, Zhifeng; ROBERTS, Adam; BOSMA, Maarten; ZHAO, Vincent; ZHOU, Yanqi; CHANG, Chung-Ching; KRIVOKON, Igor; RUSCH, Will; PICKETT, Marc; SRINIVASAN, Pranesh; MAN, Laichee; MEIER-HELLSTERN, Kathleen; MORRIS, Meredith Ringel; DOSHI, Tulsee; SANTOS, Renelito Delos; DUKE, Toju; SORAKER, Johnny; ZEVENBERGEN, Ben; PRABHAKARAN, Vinodkumar; DIAZ, Mark; HUTCHINSON, Ben; OLSON, Kristen; MOLINA, Alejandra; HOFFMAN-JOHN, Erin; LEE, Josh; AROYO, Lora; RAJAKUMAR, Ravi; BUTRYNA, Alena; LAMM, Matthew; KUZMINA, Viktoriya; FENTON, Joe; COHEN, Aaron; BERNSTEIN, Rachel; KURZWEIL, Ray; AGUERA-ARCAS, Blaise; CUI, Claire; CROAK, Marian; CHI, Ed; LE, Quoc, 2022. *LaMDA: Language Models for Dialog Applications*. Available from arXiv: 2201.08239 [cs.CL].

THORNE, James; VLACHOS, Andreas; CHRISTODOULOPOULOS, Christos; MITTAL, Arpit, 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 809–819. Available from DOI: 10.18653/v1/N18-1074.

THORNE, James; VLACHOS, Andreas; COCARASCU, Oana; CHRISTODOULOPOULOS, Christos; MITTAL, Arpit, 2018. The Fact Extraction and VERification (FEVER) Shared Task. In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, pp. 1–9. Available from DOI: 10.18653/v1/W18-5501.

TOUVRON, Hugo; LAVRIL, Thibaut; IZACARD, Gautier; MARTINET, Xavier; LACHAUX, Marie-Anne; LACROIX, Timothée; ROZIÈRE, Baptiste; GOYAL, Naman; HAMBRO, Eric; AZHAR, Faisal; RODRIGUEZ, Aurelien; JOULIN, Armand; GRAVE, Edouard; LAMPLE, Guillaume, 2023. *LLaMA: Open and Efficient Foundation Language Models*. Available from arXiv: 2302.13971 [cs.CL].

TRAN, Chau; BHOSALE, Shruti; CROSS, James; KOEHN, Philipp; EDUNOV, Sergey; FAN, Angela, 2021. Facebook AI's WMT21 News Translation Task Submission. In: *Proceedings of the Sixth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 205–215. Available also from: https://aclanthology.org/2021.wmt-1.19.

ULLRICH, Herbert, 2021. *Dataset for Automated Fact Checking in Czech Language*. Available also from: http://hdl.handle.net/10467/95430. MA thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

ULLRICH, Herbert; DRCHAL, Jan; RÝPAR, Martin; VINCOUROVÁ, Hana; MORAVEC, Václav, 2023. CsFEVER and CTKFacts: acquiring Czech data for fact verification. *Language Resources and Evaluation*. Available from DOI: 10.1007/s10579-023-09654-3.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Łukasz; POLOSUKHIN, Illia, 2017. Attention is All you Need. In: GUYON, I.; LUXBURG, U. Von; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (eds.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Vol. 30. Available also from: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

VIRTANEN, Pauli; GOMMERS, Ralf; OLIPHANT, Travis E.; HABERLAND, Matt; REDDY, Tyler; COURNAPEAU, David; BUROVSKI, Evgeni; PETERSON, Pearu; WECKESSER, Warren; BRIGHT, Jonathan; VAN DER WALT, Stéfan J.; BRETT, Matthew; WILSON, Joshua; MILLMAN, K. Jarrod; MAYOROV, Nikolay; NELSON, Andrew R. J.; JONES, Eric; KERN, Robert; LARSON, Eric; CAREY, C J; POLAT, İlhan; FENG, Yu; MOORE, Eric W.; VANDERPLAS, Jake; LAXALDE, Denis; PERKTOLD, Josef; CIMRMAN, Robert; HENRIKSEN, Ian; QUINTERO, E. A.; HARRIS, Charles R.; ARCHIBALD, Anne M.; RIBEIRO, Antônio H.; PEDREGOSA, Fabian; VAN MULBREGT, Paul; SCIPY 1.0 CONTRIBUTORS, 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. Vol. 17, pp. 261–272. Available from DOI: 10.1038/s41592-019-0686-2.

Appendix A
Translations

Colin Kaepernick became a starting quarterback during the 49ers 63rd season in the National Football League.
Savages was exclusively a German film.
Psych is a required course in California.
Congressional Space Medal of Honor is the highest award given only to astronauts by NASA.
The Penibaetic System is also called Sistema Penibético in Spanish.
Grease had bad reviews.
Commodore is ranked above a rear admiral.
Moscovium is a halogen.
Wolfgang Amadeus Mozart showed he was a child protege.
Moscovium's atomic symbol contains a G and at least two E's.
Benjamin Franklin had indefatigable campaigning for colonial unity and was regarded.
Her stars American actress Rooney Mara.
The urban center Kazan is outside of Russia.
Wilt Chamberlain averaged at least 30 points and 20 rebounds in a game.
Syd Barrett contributed to an album.
Lamniformes include the great white shark.
The iPhone 5C replaced the iPhone 5.
Buddy Holly's style was unaffected by rhythm and blues acts.
Demi Lovato is an actress.
A popular book series provided the basis for The Vampire Diaries.

Table A.1: FEVER claims sample

Colin Kaepernick se stal začínajícím quarterbackem během 63. sezóny 49ers v Národní fotbalové lize.
Divoši byl výhradně německý film.
Psychologie je v Kalifornii povinný předmět.
Congressional Space Medal of Honor je nejvyšší ocenění, které NASA uděluje pouze astronautům.
Penibetic systém je také nazýván Sistema Penibético ve španělštině.
Pomáda měla špatné recenze.
Komodor má hodnost vyšší než kontradmirál.
Moscovium je halogen.
Wolfgang Amadeus Mozart ukázal, že byl dětským chráněncem.
Symbol atomu moscovia obsahuje G a nejméně dvě E.
Benjamin Franklin neúnavně bojoval za koloniální jednotu a byl uznáván.
Její hvězdou je americká herečka Rooney Mara.
Hlavní město Kazaň leží mimo Rusko.
Wilt Chamberlain měl průměr alespoň 30 bodů a 20 doskoků v zápase.
Syd Barrett přispěl na album.
Lamniformes zahrnují velkého bílého žraloka.
iPhone 5C nahradil iPhone 5.
Styl Buddyho Hollyho nebyl ovlivněn rhythm and blues.
Demi Lovato je herečka.
Populární knižní série poskytla základ pro The Vampire Diaries.

Table A.2: WMT21 En-X (Meta AI) translations

Colin Kaepernick se stal začínajícím quarterbackem během 63. sezóny 49ers v National Football League.
Savages byl výhradně německý film.
Psych je povinný kurz v Kalifornii.
Congressional Space Medal of Honor je nejvyšší ocenění, které NASA uděluje pouze astronautům.
Penibaetický systém se také ve španělštině nazývá Sistema Penibético.
Grease měl špatné recenze.
Commodore je postaven nad kontradmirálem.
Moscovium je halogen.
Wolfgang Amadeus Mozart ukázal, že byl dětským chráněncem.
Atomový symbol Moscovia obsahuje G a alespoň dvě E.
Benjamin Franklin neúnavně vedl kampaň za koloniální jednotu a byl považován.
V ní hraje americká herečka Rooney Mara.
Městské centrum Kazaň je mimo Rusko.
Wilt Chamberlain měl v průměru nejméně 30 bodů a 20 doskoků v utkání.
Syd Barrett přispěl k albu.
Mezi Lamniformes patří žralok bílý.
iPhone 5C nahradil iPhone 5.
Styl Buddyho Hollyho nebyl ovlivněn rytmy a blues.
Demi Lovato je herečka.
Populární knižní série poskytla základ pro The Vampire Diaries.

Table A.3: Google Translation API translations

Colin Kaepernick se stal rozehrávačem týmu 49ers v 63. sezóně Národní fotbalové ligy.
Savages byl výhradně německý film.
Psychologie je v Kalifornii povinný předmět.
Vesmírná medaile cti Kongresu je nejvyšší ocenění, které NASA uděluje pouze astronautům.
Penibaetický systém se ve španělštině nazývá také Sistema Penibético.
Pomáda měla špatné recenze.
Komodor je vyšší hodnost než kontradmirál.
Moskovium je halogen.
Wolfgang Amadeus Mozart ukázal, že byl dětským chráněncem.
Atomový symbol moskviče obsahuje písmeno G a nejméně dvě písmena E.
Benjamin Franklin se neúnavně zasazoval o koloniální jednotu a byl považován za.
Její hlavní hvězdou je americká herečka Rooney Mara.
Městské centrum Kazaň leží mimo Rusko.
Wilt Chamberlain dosáhl v průměru alespoň 30 bodů a 20 doskoků za zápas.
Syd Barrett se podílel na albu.
Mezi žralokovití (Lamniformes) patří velký bílý žralok.
iPhone 5C nahradil iPhone 5.
Styl Buddyho Hollyho nebyl ovlivněn rhythm and blues.
Demi Lovato je herečka.
Předlohou pro seriál Upíří deníky se stala populární knižní série.

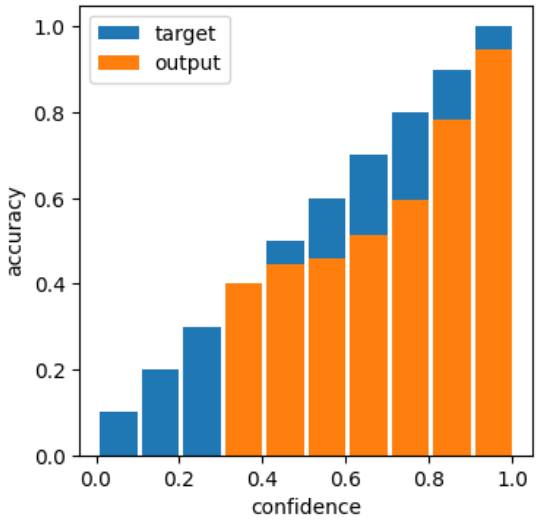
Table A.4: DeepL API translations

Appendix B
Other experiments

B.1 Temperature Scaling on Train Split

I tried to search for the answer to why is temperature scaling done only on the validation part of the dataset. Despite it remains still being unknown, I have tried to compute it and see what are the results on our dataset. The results are shown below:

(a) : Reliability diagram - development split.



(b) : Reliability diagram - development + train splits.

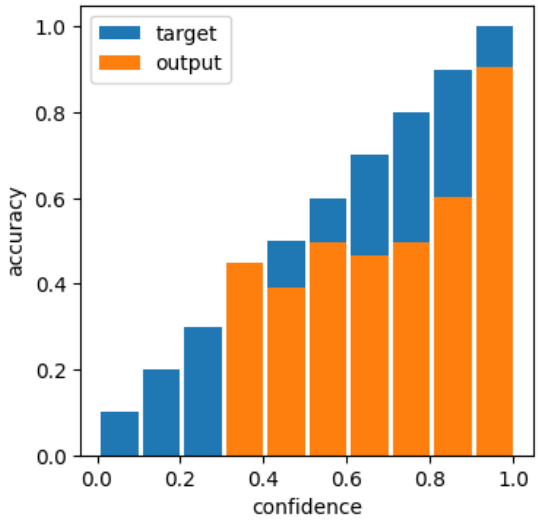


Figure B.1: Relialability diagrams.

The obtained results show us that it is better to train only development split as it was stated in the paper (Guo et al., 2017). Computed metrics are:

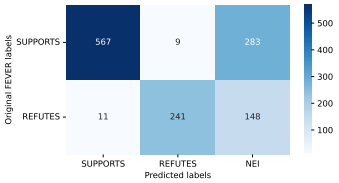
- **ECE** - Expected calibration error
  - dev: 0.0055
  - dev+train: 0.0129
- **MCE** - Maximum calibration error
  - dev: 0.1606
  - dev+train: 0.2570

The reason why it should be trained using only the development dataset only should be more explored.

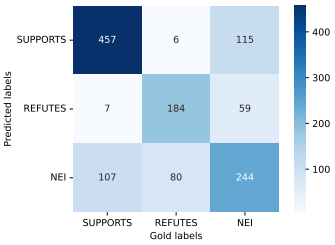
B.2 F_1 Threshold Optimization Results on Old Data

The optimal F_1 threshold was used with the NLI model trained on the noisy dataset[1] for predicting classes using the annotated data from (Ullrich, 2021). Obtained results are shown in the figures below.

(a) : FEVER against predicted labels.



(b) : Predicted labels against gold labels.



(c) : FEVER labels against gold labels.

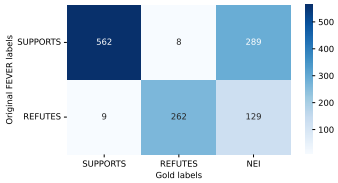


Figure B.2: Confusion matrices of the F_1 threshold filtering on old annotated data.

B.3 Data Preparation Influence on Finetuning

Table B.1 shows the influence of changes in dataset preparation. The **first version** used only evidence texts without titles (this decision was made because, in most cases, Wikipedia articles already have this title in the first sentence). The second version, F_1 **+ titles**, used a dataset with prepended titles. The third version, F_1 **+ titles, order**, added order by the list of evidence and the fourth version, F_1 **+ titles, order, Anserini**, tests Anserini hyperparameters $k1 = 0.6$ and $b = 0.5$ instead $k1 = 0.9$ and $b = 0.9$.

	Test	CTKFactsNLI
XLM-RoBERTa - F_1 first version	92.01	30.61
XLM-RoBERTa - F_1 + titles	91.74	37.44
XLM-RoBERTa - F_1 + titles, order	92.88	52.33
XLM-RoBERTa - F_1 + titles, order, Anserini	92.49	**58.08**

Table B.1: Dataset influence evaluation (F_1 score in percentage).

B.4 Validation Accuracies in NLI Finetuning

Figure B.3 shows the validation accuracies of NLI models on the datasets. Train epochs are on the horizontal axis, and the accuracies are on the vertical axis.

[1] xlm-roberta-large-squad2_bs9_ep20_wr0.4

(a): F_1 dataset.

(b): Precision dataset.

(c): 0.7 dataset.

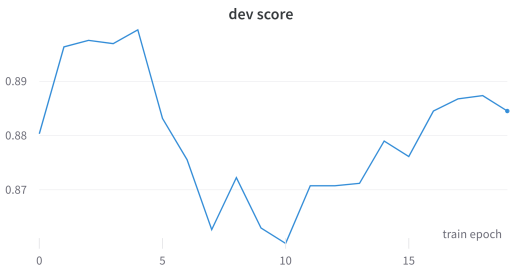


Figure B.3: Validation accuracies during finetuning of NLI models on different datasets

Appendix C
Showcase Application Screenshots

C.1 Gradio

Figure C.1: Gradio GUI - no options selected.

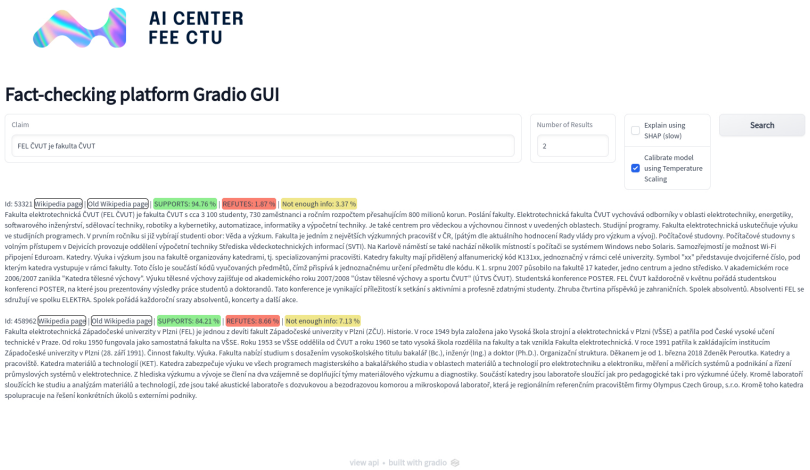


Figure C.2: Gradio GUI - only temperature scaling option selected.

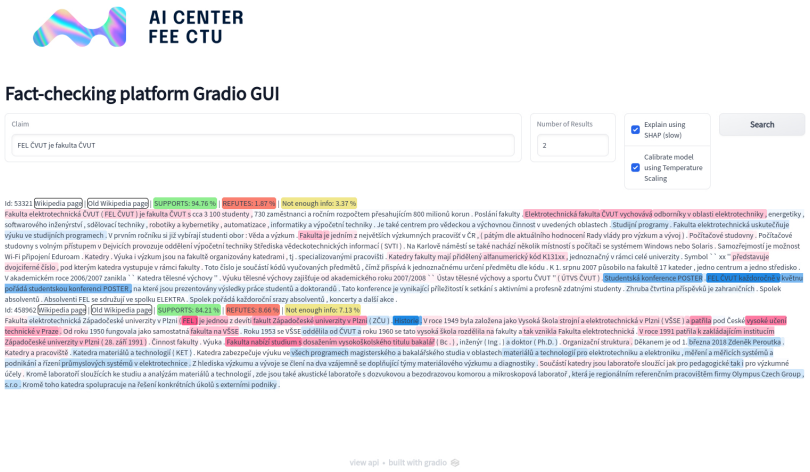


Figure C.3: Gradio GUI - temperature scaling and explain options selected.

C.2 Streamlit

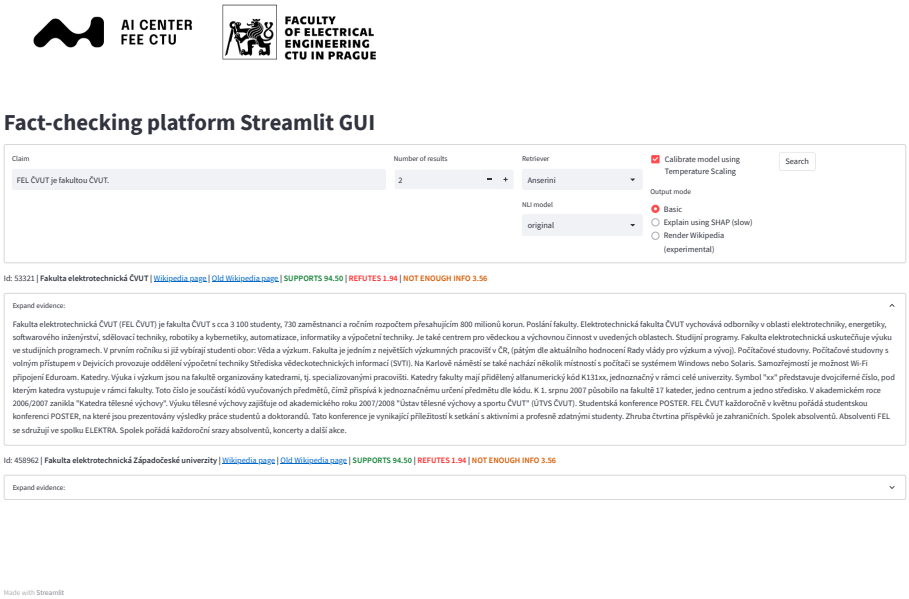


Figure C.4: Streamlit GUI - basic output mode.

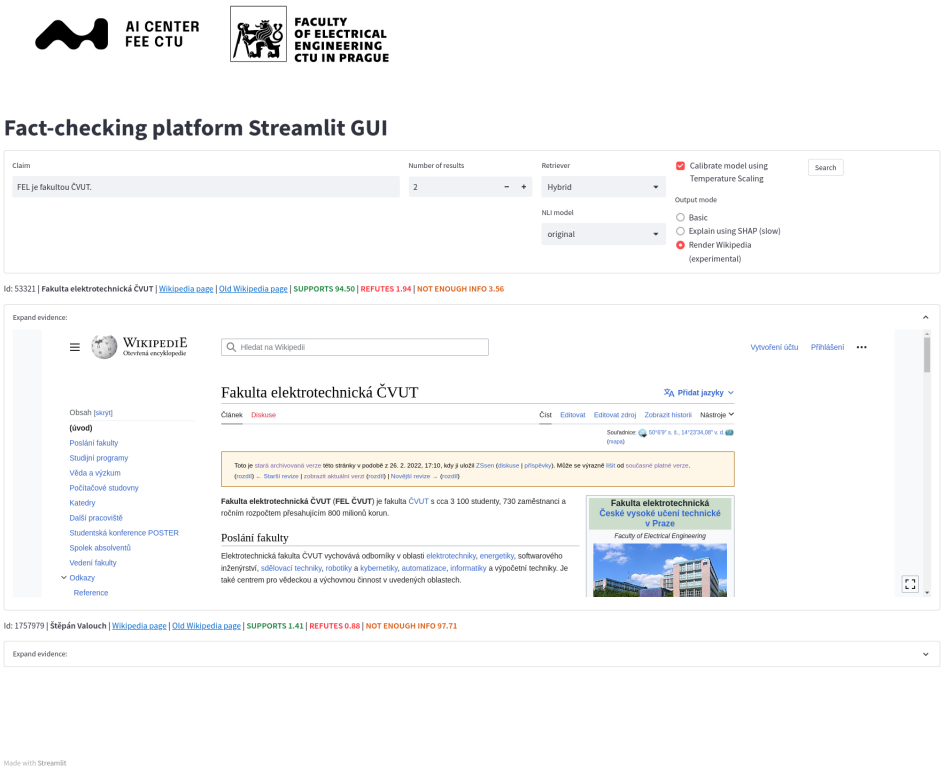


Figure C.5: Streamlit GUI - Wikipedia output mode, calibration, and Hybrid retrieval.

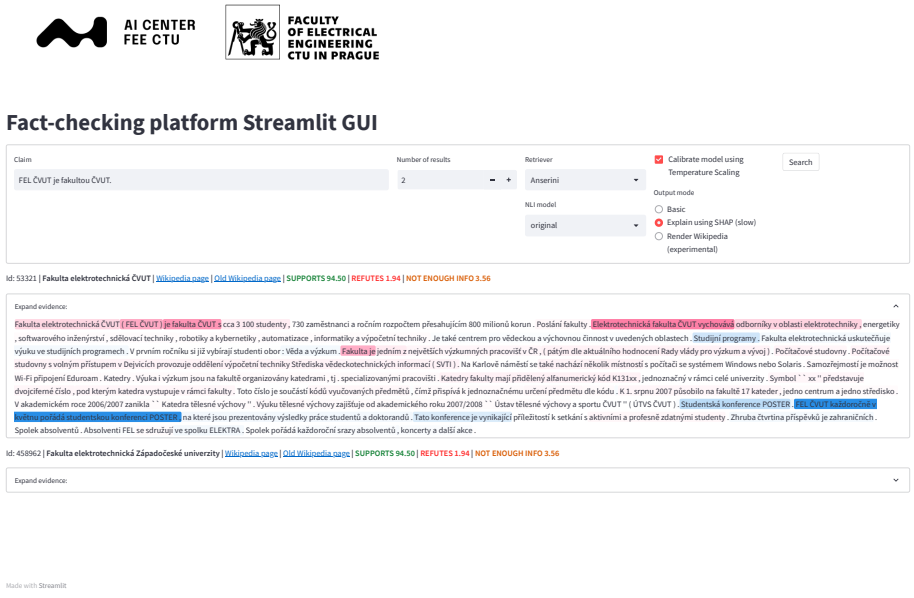


Figure C.6: Streamlit GUI - explainability output mode and calibration.

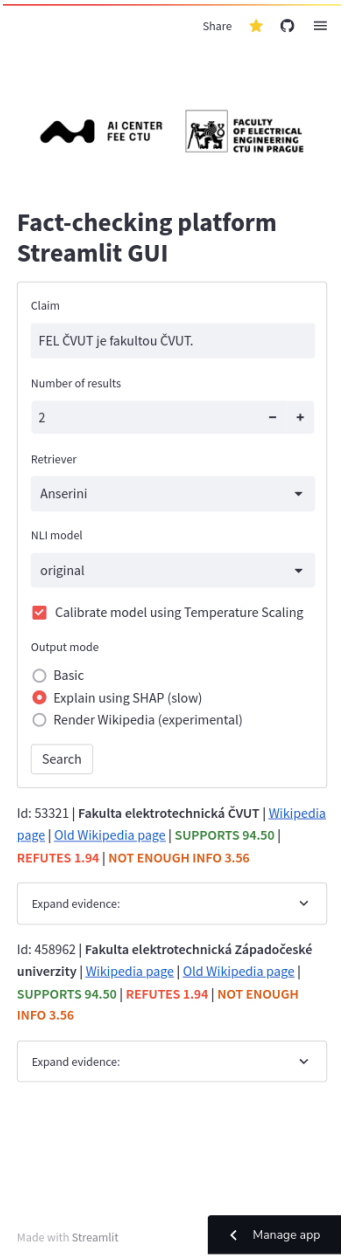


Figure C.7: Gradio GUI - responsive layout example.

Appendix D
Acronyms

NLP Natural Language Processing

NLI Natural Language Inference

RTE Recognising Textual Entailment

NEI Not Enough Info

ČTK Česká Tisková Kancelář - in english: Czech News Agency

FEVER Fact Extraction and VERification

IR Information Retrieval

TF-IDF Term Frequency - Inverse Document Frequency

BM25 Best Matching 25

BERT Bidirectional Encoder Representations from Transformers

mBERT multilingual BERT

LM Language Model

LPLM Large Pretrained Language Model

ColBERT COntextualized Late interaction over BERT

SEAL Search Engines with Autoregressive LMs

GPT Generative Pre-trained Transformer

LaMDA Language Model for Dialogue Applications

LLaMA Large Language Model Meta AI

PaLM Pathways Language Model

API Application Programming Interface

MRR Mean Reciprocal Rank

ML Machine Learning

SE Score Evidence

NSE No Score Evidence

GUI Graphical User Interface

SHAP SHapley Additive exPlanations

LIME Local Interpretable Model-agnostic Explanations

Appendix E
Repository Structure

For the codes was created a git repository on Gitlab `https://gitlab.fel.cvut.cz/factchecking/bachelor-thesis-repository-tomas-mlynar`. A snapshot of this repository is enclosed and described below.

Description of Enclosed File

- `mlynatom_bp_repository.zip` — enclosed file
  - `03_data` — scripts and notebooks related to chapter 3
    - `evaluation` — evaluation notebooks for datasets
    - `filtering` — NLI filtering notebooks
    - `translation` — translation notebooks
    - `wikipedia_dump` — scripts and notebooks for getting Wikipedia
  - `04_model_training` — scripts and notebooks related to chapter 4
    - `NLI` — NLI notebooks and scripts
      - `training` — training notebooks for NLI finetuning
      - `evaluation` — evaluation notebooks
    - `document_retrieval` — document retrieval notebooks
      - `evaluation` — evaluation notebooks
      - `indexing_tests` — notebooks for indexing and testing retrievers
  - `05_fact-checking_pipeline` — scripts and notebooks related to chapter 5
    - `evaluation` — notebooks for pipeline evaluation
  - `06_showcase_application` — scripts and notebooks related to chapter 6
    - `gradio_gui` — notebooks for Gradio GUI
    - `streamlit_gui` — scripts for Streamlit GUI
  - `other` — other useful notebooks
  - `rci_slurm` — slurm scripts for RCI cluster
  - `src` — Python modules for notebooks in other directories
    - `explanation` — explainability modules made by E. Kopecká (included for completeness)
    - `fact-check_platform` — fact-check platform related modules