Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

# NLP Methods for Automated Fact-Checking

Dissertation Thesis of
**Ing. Herbert Ullrich**

**FCHECK.FEL.CVUT.CZ**
Supervisor: **Ing. Jan Drchal, Ph.D.**
Field of study: **Informatics**
Subfield: **Natural Language Processing**
**August 2023**

# Contents

# Figures



# Tables

# Chapter 1
# Introduction

My dissertation, as well as my long-term research, centers around the field of *automated fact checking* through the means of Natural Language Processing (NLP) and its modern methods. The work consists of the analysis of the whole fact-checking process, its subdivision, and simplification into tasks that can be efficiently addressed using the current state-of-the-art NLP methods, collection of data appropriate to benchmark such tasks, delivery of example solutions and their validation against similar research in other languages and related tasks.

The main focus of mine and of our research group are the fact-checking-related tasks in the West Slavic languages (Czech, Slovak, and Polish) and secondarily in English. My contribution has so far been the collection and publication of novel datasets for the fact-checking task and its subroutines, models trained for the tasks and their debate, including the ongoing establishment of metrics that would explainably rate the model success and error rates in terms close to the human notion of *facticity* (which proves to be a challenge on its own, requiring another round of novel research [Koto et al., 2020, Wright et al., 2022]).

My doctoral aim is to cover every step on the path from gathering a factual claim – for example, extracting it from a political debate – to predicting its veracity verdict and justifying it rigorously with hard data. With the recent boom in NLP beginning with the advent of transformer networks and later the Large Language Models (LLMs) [Zhao et al., 2023], few-shot learning [Brown et al., 2020a] and prompting [Liu et al., 2023a] a significant part of the research is and has to be an appropriate and timely adoption of new ever-evolvingstate-of-the-art NLP solutions, based on well-designed studies in our specific context.

Overall, my agenda is to follow up on my published research on fact-checking in Czech with methods that reiterate on our results in other languages and evolving our previous methodology based on transformer *pre-training & fine-tuning* paradigm to a computationally feasible design based on LLMs, which are already exhibiting superiority tasks similar to ours [Chen et al., 2023] in English.

My recent focus within the whole grand fact-checking scheme is the step of *claim generation*, which I aim to establish among the other commonly benchmarked NLP tasks within the scientific community, adjacent to that of *abstractive summarization*. To benchmark the task, one would need a set of metrics that properly reflect phenomena such as *model hallucinations* – a common problem of modern-day LLMs [Ji et al., 2023]. As the exact word-level metrics for NLP generative tasks do not correlate well with human judgement [Zhang* et al., 2020] and model-based metrics are hard to explain, my research also focuses on delivery of a set of human-understandable model-based metrics.

This study aims to show the directions I am taking to address these challenges, the reasoning behind them, my research questions, and the current results that motivated


**Figure 1.1:** A real-world example of fact checking done by `https://politifact.org`

them.

## 1.1 Motivation

The spread of misinformation in the online space has a growing influence on the Czech public [STEM, 2021]. It has been shown to influence people's behaviour on the social networks [Lazer et al., 2018] as well as their decisions in elections [Allcott and Gentzkow, 2017], and real-world reasoning, which has shown increasingly harmful during the COVID-19 pandemic [Barua et al., 2020] and the Russo-Ukrainian war [Stănescu, 2022].

The recent advances in artificial intelligence have unintendedly contributed to the spread of misinformation on social media [Buchanan and Benson, 2019], as well as they hold a large potential for the false content generation [Sebastian, 2023].

Recent research has shown promising results [Thorne et al., 2019] in false claim detection for data in English, using a trusted knowledge base of true claims (for research purposes typically fixed to the corpus of Wikipedia articles), mimicking the *fact-checking* efforts in journalism.

Fact-checking (Figure 1.1) is a process of matching every information within a *factual claim* to its *evidence* (or *disproof*) in trusted data sources to infer the claim veracity and verifiability. In exchange, if the trusted *knowledge base* contains a set of “ground truths” sufficient to fully infer the original claim or its negation, the claim is labeled as **supported** or **refuted**, respectively. If no such *evidence set* can be found, the claim is marked as **unverifiable**[1].

## 1.2 Automated Fact Checking

Despite the existence of end-to-end fact-checking services, such as `politifact.org` or `demagog.cz`, the human-powered approach shows weaknesses in its scalability. By design,

[1]Hereinafter labeled as `NOT ENOUGH INFO`, in accordance to related research.

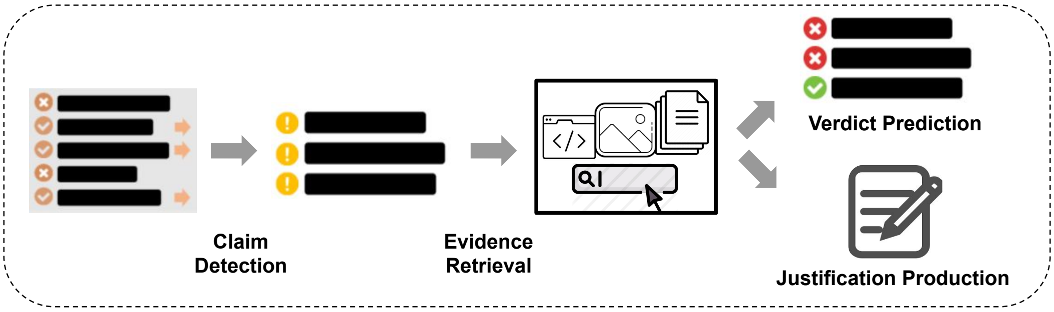


**Figure 1.2:** Automated fact-checking pipeline, reprinted from [Guo et al., 2022]

the process of finding an exhaustive set of evidence that decides the claim veracity is much slower than that of generating false or misguiding claims. Therefore, efforts have been made to move part of the load to a computer program that can run without supervision.

The common research goal is a fact verification tool that would, given a claim, semantically search the provided knowledge base (stored, for example, as a *corpus* of some natural language), propose a set of evidence (e.g., $k$ semantically nearest paragraphs of the corpus) and suggest the final verdict (Figure 1.2) [Guo et al., 2022]. This would reduce the fact-checker's workload to mere adjustments of the proposed result and correction of mistakes on the computer side.

The goals of the ongoing efforts of FactCheck team at AIC CTU are to explore and adapt the state-of-the-art methods used for fact verification or similar tasks in other languages, curate appropriate datasets for it and propose strong systems for such a task in Czech.

## 1.3 A word on the Transformers

For the past six years, the state-of-the-art solution for nearly every Natural Language Processing task is based on the concept of *transformer networks* or, simply, *Transformers*. This has been a major breakthrough in the field by [Vaswani et al., 2017], giving birth to the famous models such as Google's BERT encoder [Devlin et al., 2019a] and its descendants, or the OpenAI's GPT-3 decoder [Brown et al., 2020a] and GPT-4 [OpenAI, 2023a] that are used in the booming online AI service ChatGPT[2].

In our proposed methods, we use Transformers in every step of the fact verification pipeline. Therefore, we would like to introduce this concept to our readers to begin with.

Transformer is a neural model for *sequence-to-sequence* tasks, which, similarly, e.g., to the *LSTM-Networks* [Cheng et al., 2016], uses the Encoder–Decoder architecture. Its main point is that of using solely the *self-attention* mechanism to represent its input and output instead of any sequence-aligned recurrence [Vaswani et al., 2017].

In essence, the *self-attention* (also known as the *intra-attention*) transforms every input vector to a weighted sum of the vectors in its neighbourhood, weighted by their *relatedness* to the input. One could illustrate this on the *euphony* in music, where every tone of a song relates to all of the precedent and successive ones, to some more than to others.

The full Transformer architecture is depicted in Figure 1.3.

[2] `https://chat.openai.com`

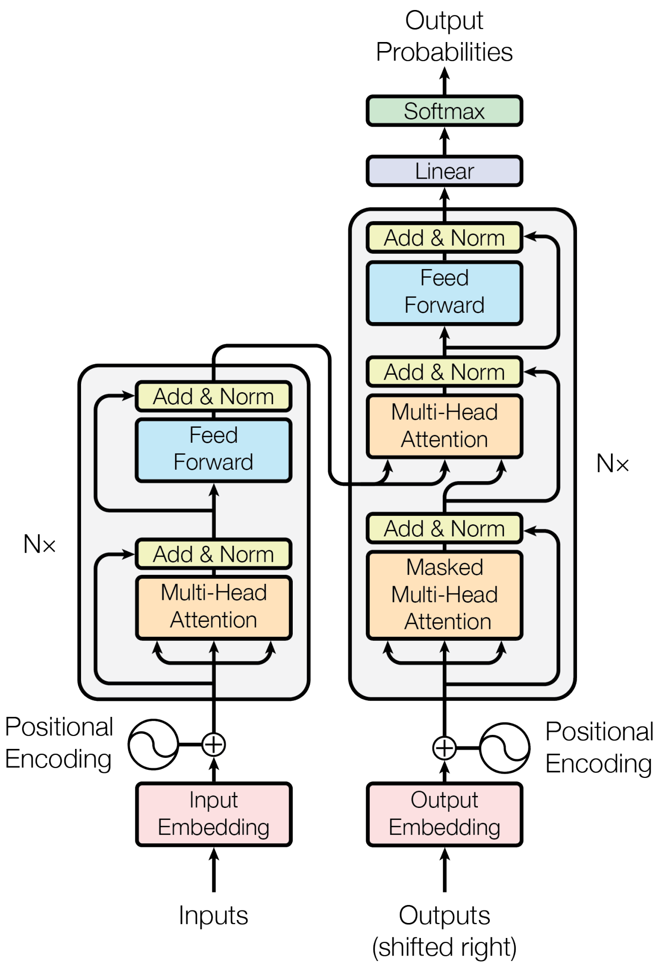


**Figure 1.3:** Transformer model architecture, reprinted from [Vaswani et al., 2017]

## 1.4 Dissertation outline

- **Chapter 1** introduces the dissertation topic, motivates the research, sets up our challenges for future research
- **Chapter 2** examines the most relevant research in the field and tries to highlight the recent paradigm shift from models trained for a single task to single large models that perform well in everything
- **Chapter 3** explains our current contributions to the field of automated fact-checking and NLP in Czech
- **Chapter 4** describes our plan for the dissertation and justifies the directions we are taking
- Finally, **Chapter 7** concludes the study with a wrapup of its findings

# Chapter 2
# State of the Art

This chapter will first describe the originally popular models for general NLP, such as BERT and the recent paradigm shift from *pretrain + finetune* transfer learning framework popular since the original [Devlin et al., 2019a] paper to the currently booming LLMs, which often outperform the smaller models even without the fine-tuning step [OpenAI, 2023a, Touvron et al., 2023a, Vicuna, 2023]. We will then take a look at the performance optimization methods that enable training multi-billion parameter pre-trained models on a set of task-specific data on a single GPU and their potential for our research.

To show how it relates to our main topics, we will introduce currently published approaches for the automated fact-checking task, efforts related to claim generation, and evaluation of NLP model outputs.

## 2.1 Pretrain + Finetune

For the last decade, the *pretrain-finetune* paradigm has been a cornerstone in Natural Language Processing (NLP). It has significantly shaped the development of modern NLP models. Its use in NLP can be traced back to the advent of neural networks and deep learning in the early 2010s. Initially, researchers pre-trained word embeddings using methods like Word2Vec [Mikolov et al., 2013] and GloVe [Pennington et al., 2014], which captured semantic relationships among words and then tweaked the general-task models for various related tasks.

### 2.1.1 BERT and derivatives

The *pretrain-finetune* paradigm truly rose to fame with the introduction of transformer-based models, particularly the revolutionary BERT (Bidirectional Encoder Representations from Transformers) in 2018. BERT [Devlin et al., 2019a] demonstrated the power of pretraining large-scale language models on massive text corpora using an easy-to-automate general task such as *Masked Language Modeling*, or *Next Sentence Prediction*, followed by fine-tuning on specific downstream tasks using smaller, harder-to-obtain data. This approach achieved state-of-the-art results across various NLP benchmarks. Subsequently, numerous variations of pre-trained models like GPT (Generative Pre-trained Transformer) and RoBERTa emerged, each refining the pretrain-finetune paradigm to improve language understanding, generation, and transfer learning capabilities.

Importantly, BERT's success inspired many publications in training similar transformer models, varying in the definition of the general pre-training task, model size, architecture training corpus

- In Czech language, monolingual models CZERT [Sido et al., 2021], FERNET [Lehečka and Švec, 2021], RobeCzech [Straka et al., 2021], and small-e-czech [Kocián et al., 2021] are available for further finetuning
- In Polish, HerBERT [Mroczkowski et al., 2021] achieved state-of-the-art in multiple tasks in 2021
- In Slovak, SlovakBERT [Pikuliak et al., 2021] was released by KInIT and Gerulata
- A multitude of multilingual models, such as M-BERT or XLM-RoBERTA [Conneau et al., 2019] were pre-trained on data in all three of these languages (and many others), proving that the large transformers can capture a notion of semantics and relations between pieces of text even *without* the convenient constriction of a single language

## 2.2 Few-shot and Zero-shot learning

The ever-growing (sometimes billions of parameters in size) transformer models have not only demonstrated superior performance on benchmark datasets but have also shown remarkable zero-shot and few-shot learning abilities, where they can perform tasks with minimal or no task-specific training data [Brown et al., 2020a].

Few-shot learning refers to the capability of a model to perform a task when provided with only a limited amount of labeled examples. Zero-shot learning takes this concept a step further by enabling models to tackle tasks they have never seen during training. The integration of these learning paradigms into large language models like GPT-3 and subsequent iterations has spread the NLP hype even further. By utilizing a prompt or a few examples, these models can quickly adapt to new tasks, making them highly versatile, adaptable, and usable to the general public.

### 2.2.1 OpenAI LLMs: GPT-3 and GPT-4

In 2020, the few-shot learning was exhibited on GPT3 – a 175B-parameter autoregressive model trained by [Brown et al., 2020a]. The model was trained on the task of generating text based on user's and its own previous outputs. The training procedure and data[1] is thoroughly described in the publication. However, it is prohibitively costly for most labs to reproduce or even fine-tune at such a scale.

In the fall of 2022, GPT-3 became widely popular thanks to its ChatGPT[2] fine-tune and demonstration app, which puts the user in the role of *prompter*, texting back and forth with an LLM that predicts the most fitting reply to each conversation.

With the arrival of GPT-4, the ChatGPT was already massively famous, and the new model already shipped with a paid-service business scheme no longer publishing the training data, tasks, or even model size [OpenAI, 2023a].

## 2.3 Open source LLMs

This puts the research community in an awkward position, as the GPT-4 achieves state-of-the-art in numerous NLP benchmarks [OpenAI, 2023a, Liu et al., 2023b], but is designed

[1] A mixture of crawled websites, books, and Wikipedia.
[2] `https://chat.openai.com`

not to be used in any way other than as a black box, making the derived research rigorosity and reproducibility disputable.

From the prediction times, OpenAI claims, and general trends in NLP, there are also reasons to believe that GPT-4 is orders of magnitude larger than already wasteful GPT-3. This motivates an uptick in research of other LLMs that would be able to operate on a smaller scale with similar results, using a peer-reviewed architecture, training scheme, and data that is available in open source.

### 2.3.1 LLaMA-2 and derivatives

A popular foundational LLM to compete with the GPT family has become the LLaMA [Touvron et al., 2023a] from Meta research. LLaMA was trained on about 5TB of publicly available textual data[3] mainly in English.

It comes in various sizes between 7B and 65B parameters, achieving a SOTA among open-source solvers in various tasks and an unmatched performance in the field of single-GPU (7B and 13B) model sizes. LLaMA proceeds to be used as a goto base model for a number of successful open-source chatbots such as Alpaca [Taori et al., 2023], Vicuna [Vicuna, 2023], and OpenAssistant [Köpf et al., 2023].

The pre-trained LLaMA weights are, however, published under a restrictive license that prohibits republishing the model weights even after tuning its parameters, which limits its fine-tuners to publishing delta- or xor-weights that can not be properly used without Meta's permission.

LLaMA-2 [Touvron et al., 2023b] addresses this inconvenience (as well as delivers its own take on the *chatbot* task), yielding an ideal strong base model for experimentation with any NLP task in 7B, 13B, and 70B sizes. The only obstacle left in the way is the computational cost of fine-tuning across so many parameters.

### 2.3.2 LoRA and other optimization

To be able to fine-tune multi-billion-parameter models such as LLaMA-2 [Touvron et al., 2023b] on a single TPU, successful approaches have been published to dramatically cut down the training expenses. Parameter-efficient fine-tuning (PEFT) [Liu et al., 2022a] proposes approaches to only fine-tune *a few* weights as opposed to the whole neural network, reducing the number of trainable parameters by orders of magnitude. Low-Rank Adaptation of Large Language Models (LoRA) [Hu et al., 2021] does so by freezing the pre-trained model weights and injecting trainable rank decomposition matrices into each layer of Transformer architecture.

Quantization, which cuts the costs of working with 32- or 16-bit float parameters and opting for data types of bitsize as small as 4, also proves to be a powerful tool for LLM finetuning performance optimization [Dettmers et al., 2023]. Quantized QLoRA takes LLaMA and finetunes it into a Guanaco model family, which outperforms all previous openly released LLMs on Vicuna benchmark [Dettmers et al., 2023] and achieves 99.3% of the ChatGPT's performance on it while only requiring 24 hours on a single GPU.

As per an alleged leaked Google memo [Patel and Ahmad, 2023], this could put the future state of the art in NLP disciplines back into the hands of open source and public research, not giving any of the big tech companies a "moat" advantage.

[3] To be specific, LLaMA was trained using an autoregressive language modeling task on a mixture of English CommonCrawl Corpus, C4 [Raffel et al., 2019], Github, Wikipedia, Gutenberg Project, Books3 corpus, ArXiv and Stack Exchange

Either way, it goes to show that the open-source LLMs have a promising future in NLP and will be indismissible as an approach for the NLP task of *Automated fact checking*.

## 2.4 Fact checking approaches

Back in the late 2010s, the misinformation and its spread in the era of the internet and social media became a discussed topic in the Western world, with multiple institutions such as the European Council marking it a severe threat to democracy and national safety [Wardle and Derakhshan, 2017]. The public attention and maturation of appropriate technologies motivated numerous efforts in business and academia to tackle the challenge. Among other events, a Fake News Challenge occurred in 2017 [Pomerlau and Rao, 2017] exploring the uses of technologies in the field and applying, for example, the LSTMs to detect stances among textual data [Hanselowski et al., 2018].

### 2.4.1 FEVER and followups

Soon, standard tasks began to be formulated and data collected. The FEVER (Fact Extraction and VERification) [Thorne et al., 2018a] dataset and shared task became prominent in natural language processing research. Relatively early on, it formalized the task as a two-step problem:

1. Retrieving information within a structured corpus to fact-check a given claim (this resembles a standard NLP problem called *information retrieval* – IR)
2. Classifying the inference relation between retrieved information and claim as one of:
   a. **supports** – information semantically implies the claim
   b. **refutes** – information semantically implies the negation of the claim
   c. **not enough info** otherwise

   This classification task became known as *natural language inference* and mostly replaced the previous binary classification NLP task of *recognizing textual entailment* (RTE)

The FEVER dataset was a collection of 185K human-annotated claims, their veracity labels, and sets of evidence from a structured corpus that sufficed to justify the labels. The corpus of choice was a 2017 English Wikipedia structured into articles due to its reasonable size, informational richness, and open license.[4]

FEVER yields an interesting benchmark with statistically quantifiable model success, motivated multiple well-performing public solutions [Thorne et al., 2018c, Thorne et al., 2019], gives insights into the complexities of automated fact-checking task, and strong baselines for research in the field. The data was later enriched by contrastive evidence in VitaminC [Schuster et al., 2021] and by reasoning over tabular data in FEVEROUS [Aly et al., 2021].

To date, it keeps being a reference point in automated fact-checking research despite its limitations, such as its requirement for a fixed knowledge base and "atomicity"[5] of claims.

[4]Is Wikipedia a trustworthy informational canon, though? No, it is not supposed to – FEVER states that it is crucial to always maintain that the fact-checking classifiers only classify *with respect to* data, and their reliability goes only as far as that of the underlying knowledge corpus. Therefore, *supports* does not directly translate to *true*, nor *refutes* to *false*

[5]See section 4.1.2.3

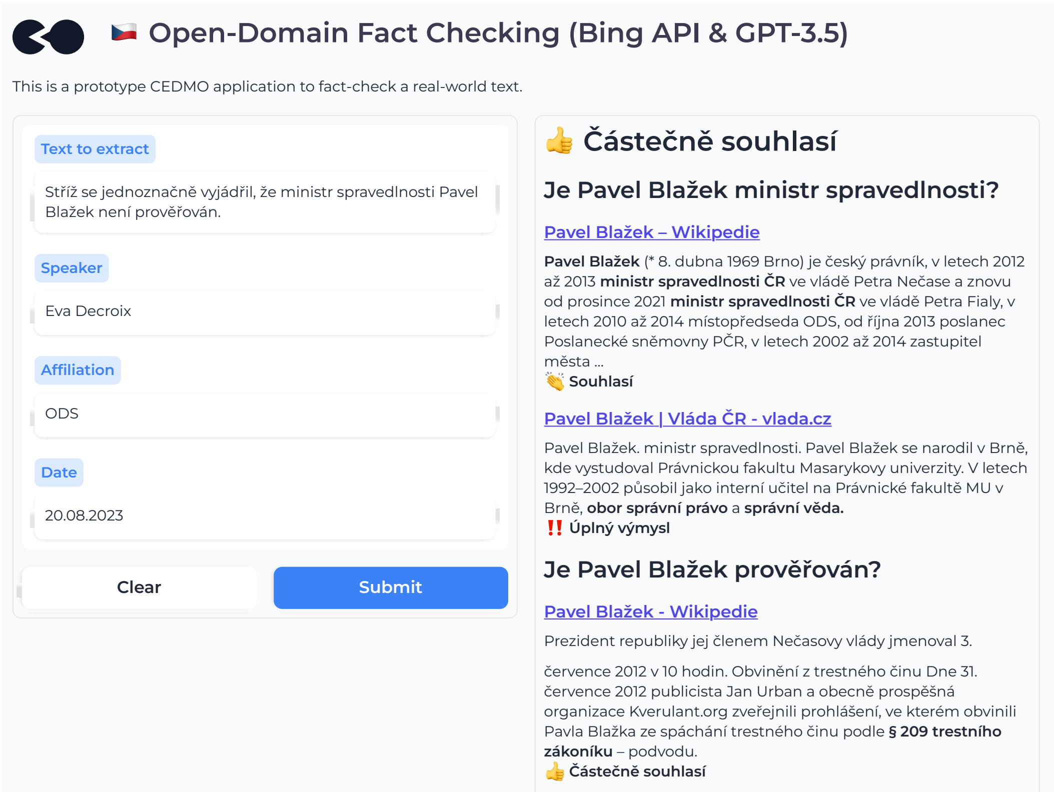


**Figure 2.1:** Proof-of-concept Czech fact-checking based on live-internet search (Bing API) and LLM prompting, based on the proposals of [Chen et al., 2023] in Czech, using a real-world claim that was fact-checked by `demagog.cz` in June 2023

### 2.4.2 Open-domain fact-checking

Due to these limitations, some researchers consider the scheme from FEVER an oversimplification – the real politics' claims to be fact-checked by journalists often consist of long syntactical structures, combine information together in a non-trivial manner and often require the most up-to-date evidence.

"Complex Claim Verification with Evidence Retrieved in the Wild" [Chen et al., 2023] proposes a different scheme that overcomes these shortcomings:

1. Arbitrarily complex claim is decomposed into a set of yes/no questions
2. An open-domain search (Bing is proposed in the paper) fetches several evidence documents for each question
3. A claim-focused summary is extracted from each document
4. A veracity classifier goes through each pair of evidence and question, ranging from "faithful" to "completely wrong"
5. The scores are combined (all need to be "faithful" for a faithful claim. Otherwise, the severity of inaccuracies can be approximated using some averaging.

GPT-3 is used in steps 1, 3, and 4 of the scheme in the prototype delivered in [Chen et al., 2023] in a few- and zero-shot fashion, with few-shot unsurprisingly coming out a little better. The scheme is transducible to Czech, and Figure 2.1 shows my early experiments with my interactive reproduction of it, predictors based on Bing and GPT-3.5 (a polished version of GPT-3).

While the shift from an established FEVER framework to complex real-world claims and evidence retrieval "in the wild" feels exciting and practical, an obvious pitfall arises – anyone can publish anything on the internet, having it appear in Bing search and other crawlers alike. I argue that this might lead into a sort of a circular dependency of needing to reliably fact-check the evidence we have retrieved from the web in order to be able to build a reliable fact-checker in the first place.

Anyhow, the open-domain fact-checking idea opens a whole new range of approaches and shows the power of LLMs in fact-checking at its every step.

## 2.5 Claim generation

Another step of the fact-checking pipeline, covered by very few research publications, is the generation of the claim to be checked in the first place [Guo et al., 2022].

The current state of things is that journalists who fact-check statements within, say, a Facebook status, need to read through the whole document multiple times, formulate its factual claims from the stances and facts expressed in the text themselves, and then fact-check each separately.

What has been examined so far were, for example:

- Using Question Generation (QG) solver and converting the questions into declarative sentences to emulate more claims and more data for fact checking [Pan et al., 2021]
- Numerous CLEF CheckThat! challenges explored the task of estimating *checkworthiness* of different parts of a long text, such as lines in a political debate [Elsayed et al., 2021, Nakov et al., 2021]
- The task of extreme summarization (XSum) consists of summarizing a long body of text into a single sentence, focusing on its most relevant aspects and facts. Large datasets XSum [Narayan et al., 2018] in English and XL-Sum [Hasan et al., 2021] in 44 languages both present expertly annotated data from BBC News for it, as their article standard features a single-sentence summary at the beginning of each text.

### 2.5.1 NLP summarization benchmarking

An important caveat to note with the NLP tasks reducing longer text to shorter text – such as summarization or claim extraction – is that the standard automatic metrics such as ROUGE [Lin, 2004] and METEOR [Banerjee and Lavie, 2005] only focus on the *content selection* aspect of tasks, based on a word-by-word overlap and were designed to use on multiple gold summaries per input, which are not often provided with modern large-scale datasets. [NLP-Progress, 2023, Zhang* et al., 2020, Zha et al., 2023]

These serious limitations make it questionable for anyone to claim state-of-the-art on these tasks and motivate research for new metrics to cover all the important aspects of claim generation and do so in correlation with expert human judgment.

This will be the topic of section 4.2, which also introduces the state-of-the-art research we are working with to arrive to a valid set of benchmarks.

# Chapter 3
# Current contribution

*We have collected novel data for the fact-checking task in our application context, emulated and scraped unavailable datasets, making them public or readying them for doing so, we have established numerous state-of-the-art models, and we are currently working on establishing the topic of claim generation as a summarization-related NLP task.*

## 3.1 Datasets

Having the automated fact-checking scheme established in chapter 2, every machine-learning solution must start with the choice or collection of appropriate training data. Due to the novelty of the task in Czech and other West Slavic languages, I explored a multitude of ways to acquire such data, many of them resulting in a publicly available dataset in our Huggingface repository [1], beginning to be reused by others.

### 3.1.1 CsFEVER

An early "temporary benchmark" for our endeavors in adapting the FEVER [Thorne et al., 2018a] task for the Czech context was the CsFEVER [Ullrich et al., 2023a] dataset.

In [Ullrich, 2021], I have proposed a simple FEVER data transduction scheme that can be simplified as follows:

1. Each FEVER claim is translated using a Machine Translator
2. Evidence from English Wikipedia is not translated using MT, but mapped onto its Czech-Wikipedia counterpart using the publicly available Wikidata[2]
3. Data with any loss in evidence due to step 2. is discarded

This design was relatively cheap to compute (as translating the whole 2017 Wikipedia corpus would have been a long and wasteful computation), delivering an open-license dataset of 127K claims, their labels, and evidence justifications. My hope was, as both the 2017 EnWiki and our 2020 CsWiki corpus only featured the first paragraph (abstract) of each article, a document-level alignment could be assumed – both the Czech and English text always summarize the basic facts about the same entity.

This showed to be only partly true as a later human annotation on a 1% sample of CsFEVER data showed that about a third of data exhibits some levels of noise, mostly introduced during dataset translation [Ullrich et al., 2023a].

[1] https://huggingface.co/ctu-aic

[2] Used, for example, for showing the "see this article in other languages" suggestions in Wikipedia sidebar

While noisy, the CsFEVER data still got its use in the training of the information retrieval schemes of [Rýpar, 2021, Gažo, 2021, Ullrich et al., 2023a] used to this day and is openly available[3] under a CC license.

My research on it also motivated the creation of an inference-only version of the dataset, which does not support the Information Retrieval task and, therefore, does not require the mapping of evidence into a live version of Wikipedia. Therefore, only the EnWiki *excerpts* needed to build evidence can be translated, bringing down the computational difficulty and enabling me to deliver a dataset without the transduction noise called CsFEVER-NLI[4].

Another round of research CsFEVER motivated, and I supervised, was the successful thesis of [Mlynář, 2023], modernizing the data and machine-translation methods into the 2023 state of the art. [Mlynář, 2023] further experimented with methods of automated noise detection and removal, which has not shown to be an efficient way to tackle the issue of high noise in CsFEVER.

Anyhow, it delivers a partly cleaned version of it[5] and motivates future research to generate such data differently, using a claim generation scheme like that from [Pan et al., 2021].

### 3.1.2 FCheck annotations platform

The imperfections in translated CsFEVER data, as well as the ongoing collaboration with ČTK and the Faculty of Social Sciences, brought me to also look for ways how to hand-annotate a whole new natively Czech dataset, which would both lack the noise introduced in translation and also take the task of automated fact-checking to the next level, replacing a rigid, simple Wikipedic data with a more "real world" news report corpus of ČTK.

Figure 3.1 shows an open-source platform FCheck[6] I developed to collaborate with 316 FSV CUNI students of on a collection of novel dataset in Czech using ČTK data as a ground truth corpus.

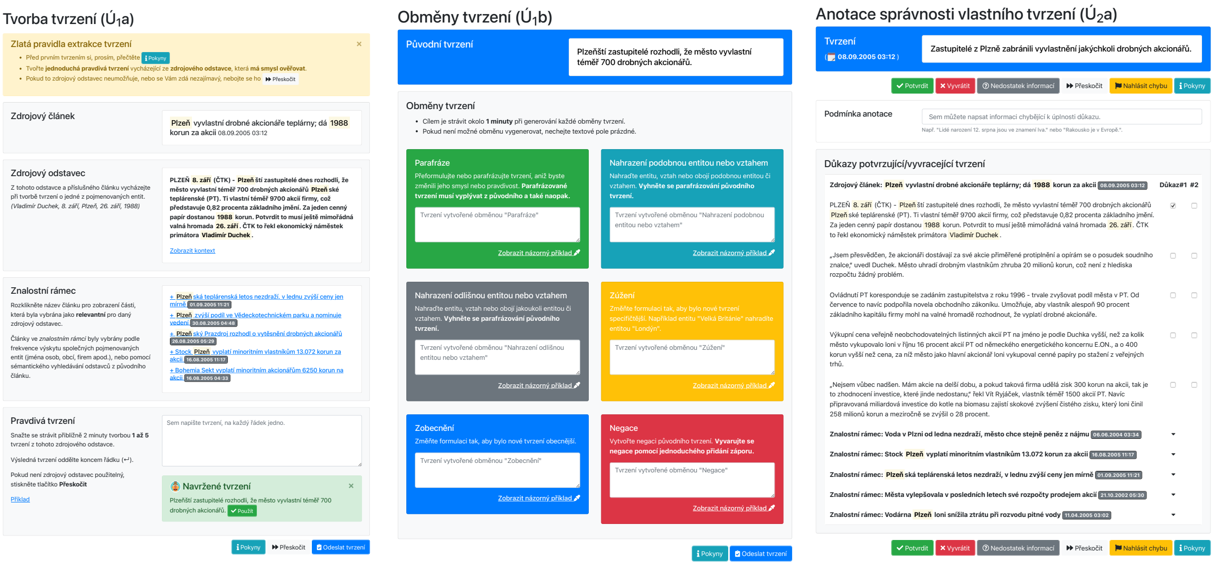


**Figure 3.1: FCheck** – a platform for fact-checking data collection developed for TAČR project; collects data for claim generation, information retrieval, and natural language inference tasks

[3] `https://huggingface.co/datasets/ctu-aic/csfever`
[4] `https://huggingface.co/datasets/ctu-aic/csfever_nli`
[5] `https://huggingface.co/datasets/ctu-aic/csfever_v2`
[6] `https://fcheck.fel.cvut.cz` (`testuser`), source at: `github.com/aic-factcheck/fcheck-annotations-platform`

We have established a 4-step annotation procedure inspired by the time-proven methodology of [Thorne et al., 2018a] where check-worthy paragraphs are first hand-picked among samples from the whole archive of ČTK's 3.3 M news reports published between 1 January 2000 and 6 March 2019. Then, the annotator is sampled such a paragraph and asked to *extract claims* from it, i.e., formulate single-sentence summaries of some facts that appear in the paragraph. This claim is always *supported* by the data, so the next phase is to perturb the claim by the annotator's world knowledge and form the claim *mutations* – substitutions of entities, generalizations, specifications, paraphrases or negations of the original claim. The mutated claim is then fact-checked by (typically) another annotator, using the ČTK data narrowed down to a reasonable number of relevant articles (in an IR sense) as *supportable*, *refutable* or *not enough info*, providing a set of evidence as a verdict justification.

The whole application is running on multiple levels – a yii-framework-powered PHP app is running the annotation interface, while a flask server in Python is running our models based on TF-IDF [Chen et al., 2017] and mBERT (section 2.1.1) for information retrieval trained among other data on the CsFEVER dataset (section 3.1.1). The models are solving the Information Retrieval task on-demand (with cache) on the proprietary ČTK corpus whenever the annotation app needs it to provide context to the fact-checker.

The scheme and its implementations are exhaustively described in [Ullrich, 2021], chapter 4, and in [Ullrich et al., 2023a], also chapter 4. Multiple "cross-annotations" were collected for each claim to measure agreement and give insights into task complexity.

### 3.1.3 CTKFacts

After completing the first year of annotation experiments, we have extracted a total of 3,116 multi-annotated claims. 47% were `SUPPORT`ed by the majority of their annotations, `REFUTES` and `NEI` labels were approximately even, the full distribution of labels is listed in Table 3.1.

| | **CTKFacts** uncleaned, balanced | | | **CTKFacts (launch)** cleaned, stratified | | |
|---|---|---|---|---|---|---|
| | SUPPORTS | REFUTES | NEI | SUPPORTS | REFUTES | NEI |
| train | 1,164 | 549 | 503 | 1,104 | 556 | 723 |
| dev | 100 | 100 | 100 | 142 | 85 | 105 |
| test | 200 | 200 | 200 | 176 | 79 | 127 |

**Table 3.1:** Label distribution in CTKFacts splits before and after cleaning. Reprinted from [Ullrich et al., 2023a]

Of all the annotated claims, 1,776, that is 57%, had at least two independent labels assigned by different annotators. I used this multiplicity to assess the quality of our data and the ambiguity of the task, as well as to propose annotation cleaning methods used to arrive at our final cleaned CTKFacts dataset.

#### Inter-annotator agreement

Due to our cross-annotation design, I had a generously sized sample of independently annotated labels in our hands. As the total number of annotators was greater than 2, and as missing observations were allowed, I have used the Krippendorff's alpha measure [Krippendorff, 1970] which is the standard for this case [Hayes and Krippendorff, 2007]. For the comparison with [Thorne et al., 2018a] and [Nørregaard and Derczynski, 2021], I also list a 4-way Fleiss' $\kappa$-agreement [Fleiss, 1971] calculated on a sample of 7.5% claims.

I have calculated the resulting Krippendorff's alpha agreement to be 56.42% and Fleiss' $\kappa$ to be 63% and interpreted this as an adequate result that testifies to the complexity of the task of news-based fact verification within a fixed knowledge scope. It also encourages a round of annotation-cleaning experiments that would exploit the number of cross-annotated claims to remove common types of noise.

#### CTKFacts publication

CTKFacts dataset was then subject to a thorough human-in-the-loop data cleaning until a 100% agreement among the data was reached, in order to remove data that contains obvious noise and reveal phenomena that lead to erroneous annotations. The full process, as well as its results, are described in [Ullrich et al., 2023a].

Ultimately, a dataset of 3.1K thoroughly cleaned data points in the form of a factual claim, its veracity label and justifications consisting of ČTK paragraphs was published in a version for Information Retrieval[7] for those who have access to the ČTK knowledge base to retrieve from, as well as in a special version for the task of Natural Language Inference[8] containing all the required ČTK excerpts we have negotiated to publish under open license for everyone to use.

The datasets have become our standard benchmark within the AIC NLP group [Semin, 2023, Mlynář, 2023] and are starting to be referred and used in others' research in the field [Štefánik et al., 2023].

### 3.1.4 Other NLP datasets in West Slavic languages

Over time, we have accumulated numerous sets of data in Czech and other Slavic languages that have previously been poorly covered or not available at all, some of which are to be referred in our future publications. For the convenience of others, most of them are already listed in our public repositories. Let us mention some significant examples:

1. We have machine-translated the most popular NLI training and benchmark datasets such as Stanford NLI [Bowman et al., 2015], Adversarial NLI [Nie et al., 2019b] and MultiNLI [Williams et al., 2018] picking a machine translator empirically for each dataset between DeepL [DeepL, 2021], Google Translate [Google, 2021] and CUBBITT [Popel et al., 2020].

   The resulting datasets are maintained at our public repositories:

   a. `https://huggingface.co/datasets/ctu-aic/snli_cs`
   b. `https://huggingface.co/datasets/ctu-aic/anli_cs`
   c. `https://huggingface.co/datasets/ctu-aic/multinli_cs`

2. For the task of claim generation we are establishing and performing in Czech, we have adapted the existing related datasets and are working with:

   a. CTKSum – `https://huggingface.co/datasets/ctu-aic/ctksum` based on source articles and extracted claims within the original CTKFacts set
   b. FEVERSum (based on FEVER Wikipedia abstract and extracted claims) – `https://huggingface.co/datasets/ctu-aic/fever-sum`

[7] `https://huggingface.co/datasets/ctu-aic/ctkfacts`
[8] `https://huggingface.co/datasets/ctu-aic/ctkfacts_nli`

c. Its DeepL translation CsFEVERSum – `https://huggingface.co/datasets/ctu-aic/csfever-sum`

d. Our reproduction of a crawled Slovak summarization dataset described by [Šuppa and Adamec, 2020] SMESum based on articles from `https://sme.sk` – `https://huggingface.co/datasets/ctu-aic/smesum`

Up until now, some of the data was restricted to private repositories, but with this study, I am publishing most of them, as I have now found the licensing to be rather relaxed. If some of the repositories the reader might be interested in would not be reachable, please request access to the `https://huggingface.co/datasets/ctu-aic` organization to be able to see into the private part of our dataset library.

## 3.2 Models

The most significant pre-trained models I have made public address two tasks – the Natural Language Inference and Claim Generation viewed as a form of Abstractive Summarization task.

### 3.2.1 Natural Language Inference

My previous work [Ullrich, 2021, Ullrich et al., 2023a] also focused on establishing a strong starting state of the art on our own datasets in the tasks of NLI. In my publications, I have tried and compared a multitude of neural networks for the tasks, ultimately arriving at the following:

- **XLM-RoBERTA-Large@XNLI@CsFEVER-NLI**, a model with 561M parameters trained on 100-language CommonCrawl corpus finetuned on multilingual XNLI [Conneau et al., 2018] inference dataset and then finetuned *again* on the CsFEVER-NLI task yields an unmatched 73.7% F1 macro score on the denoised CsFEVER-NLI inference task: `https://huggingface.co/ctu-aic/xlm-roberta-large-xnli-csfever_nli`
- **XLM-RoBERTA-Large@SQuAD2**, a model version finetuned on a Question answering SquAD2 [Rajpurkar et al., 2016] task has shown remarkable practicality in my NLI applications and after task-specific finetuning, it was able to tackle:
    1. CTKFactsNLI[9] task with 76.9% macro-F1
    2. CsFEVER[10] (noisy) task with 83.2% macro-F1
    3. The original English FEVER NLI task[11] [Thorne et al., 2018a, Nie et al., 2019a], achieving 75.9% macro-F1 and a significant superiority over previous shared task winner [Nie et al., 2019a] (which had 69.5 macro-F1 with NSMNs)

### 3.2.2 Claim generation

In my current research, I am finding appropriate configurations and data to train models for claim generation – generating a factual claim (or more) into a single sentence containing a fluent, atomic, decontextualized, and faithful claim. In section 4.1, I propose the claim

[9] `https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-ctkfacts_nli`
[10] `https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-csfever_nearestp`
[11] `https://huggingface.co/ctu-aic/xlm-roberta-large-squad2-enfever_nli`

generation as an abstractive summarization setting, and therefore, the models already have their practical use in the general task of summing up longer texts into shorter ones.

As has been shown in section 2.5.1, the NLP summarization task does not have a reliable standard benchmark that would capture all its required output qualities. Therefore, it remains questionable to claim the state of the art on any summarization task, and I proceed to present models that excel in our empirical tests and demonstrations for project stakeholders:

1. **mBART** [Liu et al., 2020] multilingual Transformer model has been finetuned by our team's [Krotil, 2022] on SumeCzech and proprietary CNC News summarization dataset on the "full text to headline" task, obtaining encouraging scores across numerous summarization metrics in Czech.
   I have taken this model a step further for the claim generation task, finetuning it on the CsFEVERSum and CTKFactsSum datasets, yielding a working model for the task.[12]

   Other experiments are being carried out with the same model finetuned on Slovak[13] and Polish[14] data.

2. **LLaMA-2** shows promising results when it comes to claim generation. I have finetuned[15] it using the QLoRA (section 2.3.2) approach, XL-Sum [Hasan et al., 2021] dataset and a concatenation-based prompting strategy [Touvron et al., 2023b], to facilitate training across the entire length of input.

All prototype models are currently being iterated with our CEDMO[16] project partners (fact-checkers from European organizations), tweaked, and future tests are being designed for them based on empirical results and questionnaires.

An application in the figure 3.2 demonstrates the single or multiple claim generation task with our LLaMA-2 or mBART models for English and Czech texts, respectively – I put it together as a GRADIO interactive application and an API. Another interactive application App Search 3.3 developed by Jan Drchal [Mlynář, 2023] demonstrates our best-performing models for the whole fact-checking tasks, integrating the XLM-RoBERTas trained on CsFEVER-NLI data.

[12] `https://huggingface.co/ctu-aic/mbart25-large-eos`
[13] `https://huggingface.co/ctu-aic/mbart-at2h-cs-smesum-2`
[14] `https://huggingface.co/ctu-aic/mbart-at2h-cs-polish-news3`
[15] `https://huggingface.co/ctu-aic/Llama-2-7b-xlsum-en`
[16] `https://cedmohub.eu`

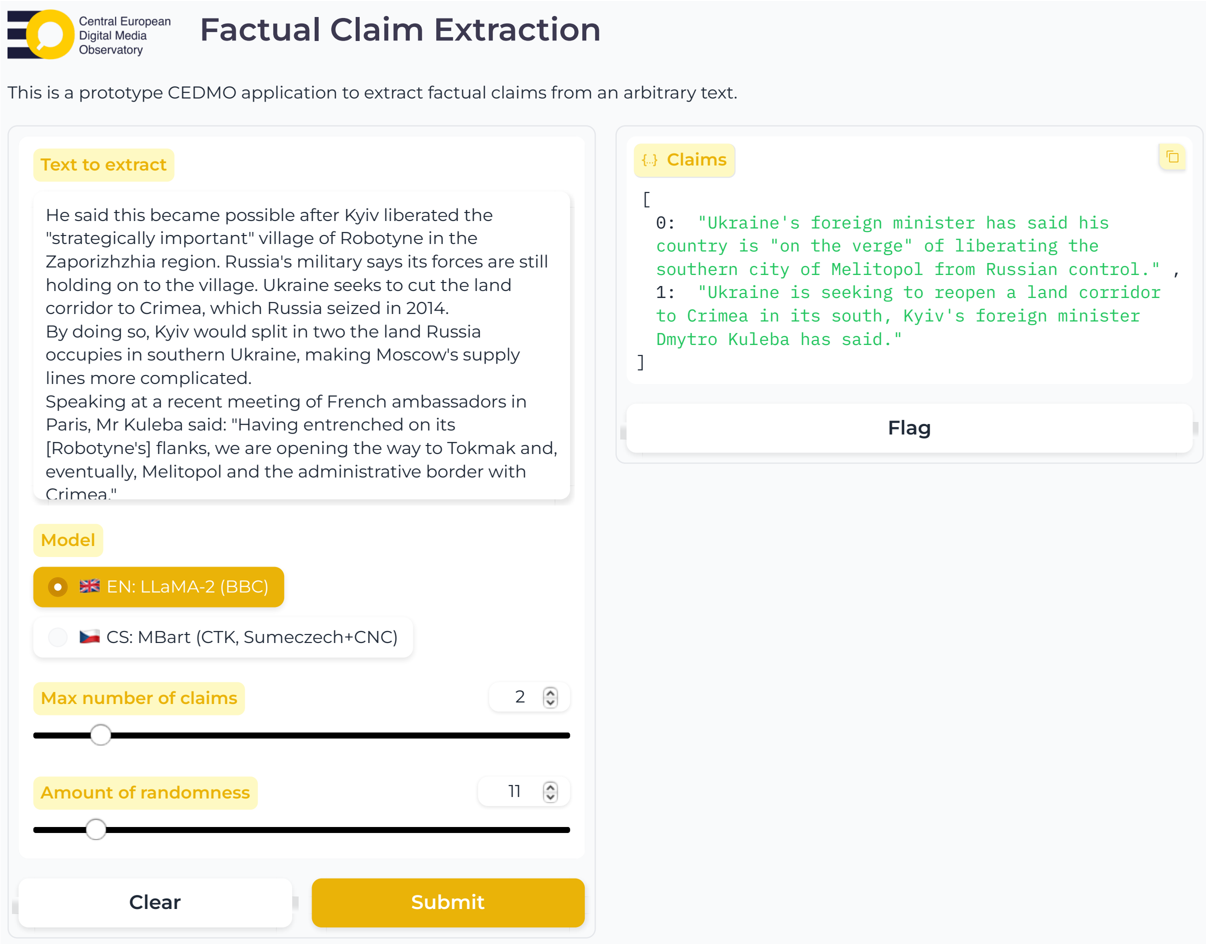


**Figure 3.2:** Factual claim extraction application done for the CEDMO project

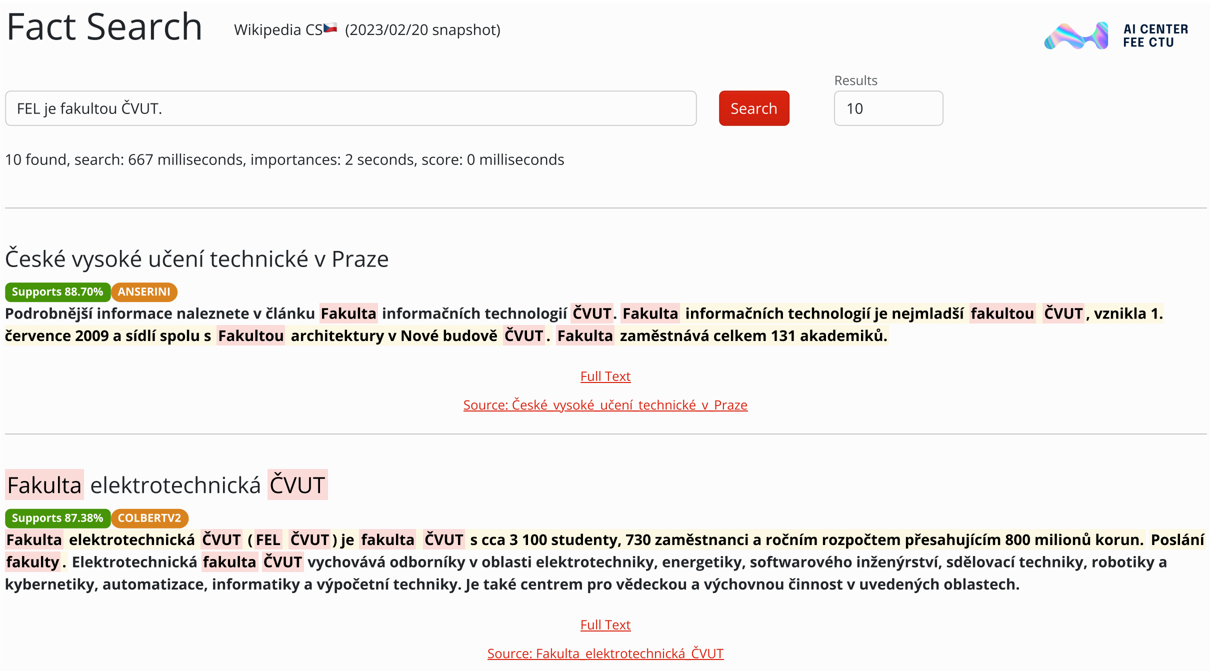


**Figure 3.3:** Automated fact-checking application “fact-search” verifying claims against Czech Wikipedia using our SOTA models

# Chapter 4
# Dissertation plan

*This chapter describes my current research agenda and its meaning for my future dissertation thesis on automated fact-checking using NLP methods, which is outlined in the last section.*

## 4.1 Automated claim generation

The article I am currently readying for submission proposes the task of *automated claim generation* as the process of extracting factual claims from a textual document. It has multiple uses in practice, such as assisting the fact-checkers and emulating data for NLP tasks like automated fact-checking and NLI.

Extracting a set of factual, atomic claims from a chunk of naturally formed text poses many challenges – what single piece of information characterizes the text best? How does one resolve the pronouns and coreferences in source text? How does one adapt the extraction scheme for different speakers and stylistic forms?

I find these problems to overlap with those of the *abstractive summarization* task, which has recently seen an advent of efficient solutions based on Transformer models [Zhang et al., 2020, Liu et al., 2022b].

The summarization scheme only requires minor tweaks – preventing it from outputting more than one sentence of output per input and training it on data appropriately chosen to promote the summarization of more than one fact when sampling different claims from the same model using the top-$k$[1] and top-$p$[2] [Holtzman et al., 2020] strategies.

An initial training data is derived from XL-Sum [Hasan et al., 2021], EnFEVER and CTKFacts, and models are trained using mBART, Pegasus, T5 [Raffel et al., 2019], and LLaMA-2 (QLoRA) architectures. Additional experiments are being carried out using the GPT Chat API.

Moving forward, the primary tasks will be a cyclic iteration of claim generation data and models, refining each part after progression in the other, and most importantly, a set of reliable metrics that are explainable and correlate with human judgment. As shown in section 2.5.1, the standard automated summarization metrics are not appropriate as a benchmark for the task.

## 4.2 Claim generation metrics

The common problem with generative tasks in NLP is that of explaining model reasoning in a human-understandable manner and troubleshooting the prediction faults, such as the

[1]Each output token is sampled from the $k$ most probable words in the dictionary

[2]Each token is sampled from the most probable words which have at most $p$ total probability mass

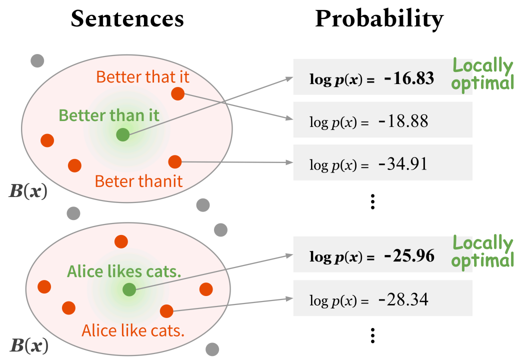


**Figure 4.1:** LM-Critic – deciding text fluency viewed as finding local optima of Language Model output probability, reprinted from [Yasunaga et al., 2021]

*model hallucination.*

For the task of claim generation, where we also face the challenge of the *relevance* of the information extracted by the model, we suggest the use of the following metrics rooted in the current research on similar topics:

1. **Fluency** – *is the claim grammatically correct and intelligible?*

   Currently, we are working with two emulations of claim fluency, challenge that is similar to a standard NLP task of Gramatical Error Detection (GED): LM-Critic (Figure 4.1) [Yasunaga et al., 2021] perturbs the claim words and characters to find local optima in output probability of its tokens, using a language model such as GPT-2 as its reference. GPTScore [Fu et al., 2023] uses prompting a LLM (such as GPT-3) to obtain a model-inferred score using few- or zero-shot learning.

   Both can be adapted for Czech and the latter is demonstrated in Figure 4.2.

2. **Decontextualization** – *can the claim be correctly interpreted without any additional context from the source document or elsewhere?*

   A common problem with machine-extracted factual claims is reusing excerpts from source documents along with inexplicable contextual pronouns ("President won't sue *them*") and relative referencing ("*Last year*, CTU had 23K students").

   [Choi et al., 2021] proposes decontextualization as a sequence-to-sequence task with two texts on input $(s, c)$ – sentence and context. T5 model [Raffel et al., 2019] is then trained on machine-generated gold data from Wikipedia to output sentence $s'$ such that the truth-conditional meaning of $s'$ in an empty context is the same as that of $s$ in $c$.

   [Mohri et al., 2023] improves upon this, altering the problem formulation to minimization of surrogate loss, rejecting with a fixed predictor, and claiming to get as close as $\sim 3\%$ away from the theoretical limit for the task.

   The approaches are reproducible using the Czech Wikipedia corpus and appropriate for further examination.

3. **Atomicity** – *does the claim describe a single entity, relation or process?*

   Claim atomicity can be checked using the Relationship Extraction methods such as LUKE [Yamada et al., 2020]. Simply put, the RE task is to identify the entities of a text (persons, institutions,...) and the relations between them (such as

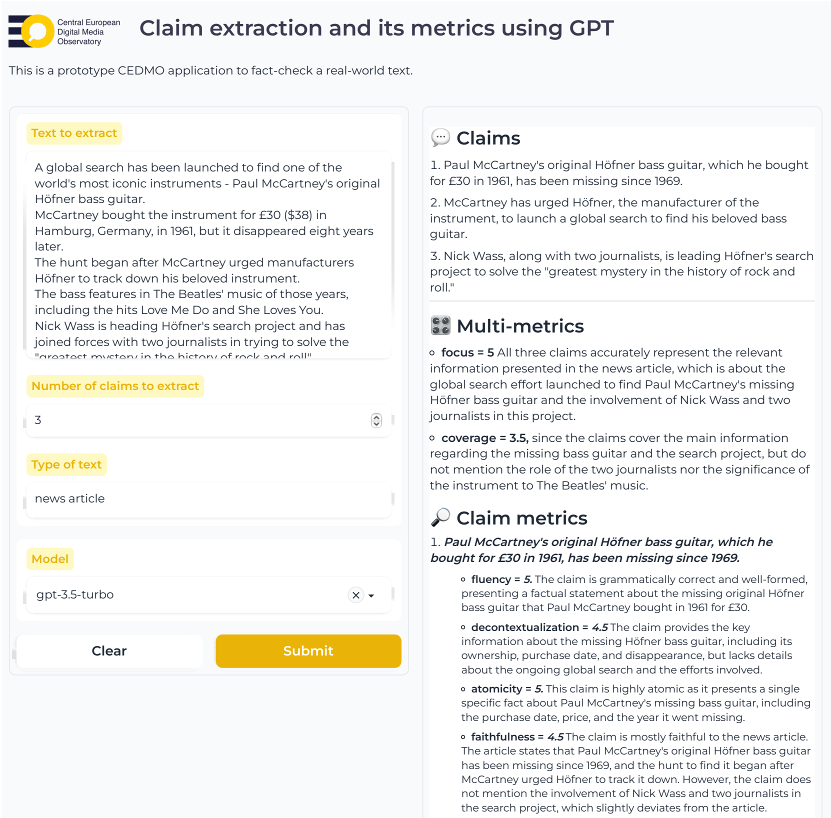


**Figure 4.2:** A self-evaluating claim generation model based on GPT-3.5-turbo and GPT-4 [OpenAI, 2023a] using the OpenAI API and a single-shot (one gold example given) approach

(*"study at"*, *Herbert*, *CTU*)). The atomicity evaluation can be converted to a RE task by attempting to extract such fact triples and mark the claim as atomic if there is at most one such triple found (after removing symmetries)

4. **Faithfulness** – *does the claim only contain information that is consistent with the source document?*

   This metric is crucial to pinpoint *model hallucinations* – parts of the claim where the model outputs stray from the information present in the source text and begin to just "make stuff up". We proceed to use two alternative metrics – a score proposed within the FFCI evaluation framework [Koto et al., 2020] as:

$$\text{AvgTop-}n_{s_j \in X, t_i \in Y'}(\text{BERTScore}(t_i, s_j))$$

   Where the AvgTop-$n$ simply averages across the top $n$ (say, 5) highest scores, $X, Y'$ are the sets of sentences in the source document and model output, respectively (so, in the claim generation scenario $|Y'| = 1$) and BERTScore [Zhang* et al., 2020] is a recently popular similarity score between two sentences that doesn't compare the texts on a verbatim level (like, e.g., ROUGE [Lin, 2004] which correlates poorly with human judgment) but expresses the sentence similarity as a sum of cosine similarities between their tokens' embeddings – this should capture semantical relations rather than the word-for-word similarity, which could be beneficial in highly inflected languages such as Czech.

   Similar metric called AlignScore was proposed in [Zha et al., 2023], looking for optimum alignment of output and input parts, in terms of a RoBERTa model [Liu et al., 2019] trained to detect inconsistencies on 4.7M training examples adapted from various tasks (inference, question answering, paraphrasing,...) and while it is relatively small (355M parameters), it outperforms metrics based on GPT-4 that is orders of magnitude larger.

Empirically, the models work encouragingly well on spotting hallucinations and inconsistencies in English, and while the transduction of BERTScore is trivial, using a Czech embedding model such as CZERT [Sido et al., 2021] or FERNET [Lehečka and Švec, 2021], reproducing the success of AlignScore will require more research and data.

5. **Focus**@$k$ – *if we generate $k$ claims using this model, what will be the proportion of gold (relevant) information among all the information listed in the generated claims?*

   The metric is analogous to the concept of *precision* in the common machine learning applications; however, its deciding gets more ambiguous in the natural language setting, where we are dealing with synonyms and an endless number of possible wordings for every piece of information.

   An elegant and functional perspective on the problem has been brought around in QAGS[3] evaluation protocol [Wang et al., 2020], where the idea is to use a Question Generation model (QG) to formulate questions in natural language based on all $k$ predicted claims. The questions are then twice answered using a Question Answering (QA) model, giving it knowledge from (i.) the predicted claims (ii.) the gold claims written by a human. The focus is then defined as the proportion of questions with the same answers extracted from the gold and predicted claims among all questions the model can generate from the predicted claims.

6. **Coverage**@$k$ – *if we generate $k$ claims using this model, what proportion of gold (relevant) information from the source text will be covered?*

   Analogous to *recall@k* in general machine learning, QAGS proposes to generate questions using gold claims and try to answer them using the predicted claims, much like in the *focus* scenario, but vice versa.

The metrics are proposed in accordance with other research on model-based evaluation of similar NLP tasks [Koto et al., 2020, Wright et al., 2022] and are to be refined upon experiments with annotators.

## 4.3 Data collection

### 4.3.1 Human-in-the-loop grading of claim generators

To validate and progress the metrics referred to in section 4.2, one needs human-annotated data for the task. I aim to use an experiment similar to that of [Wright et al., 2022], presenting annotators with ordinal scales for the claim qualities and appropriate grading for each metric conditioned by objective rules.

My research will attempt to design the experiment in a way that yields the best data, checking its validity using inter-annotator agreement and other forms of feedback and publishing the data and scheme alongside the other solutions. Collected data will be used to validate the prototype metrics from section 4.2 and propose their variations based on the findings.

[3] Pronounced “kags”, stands for “Question Answering and Generation for Summarization”

### 4.3.2 Polish dataset scraping

While Czech has its SumeCzech [Straka et al., 2018] and in Slovak, we can still reproduce the SMESum [Šuppa and Adamec, 2020] research, a large-scale single-sentence summarization dataset in Polish has yet to be established. The closest data I have found is the online news corpus [Szwoch et al., 2022] collected for the purposes of studying political polarization (and nowhere published, despite my e-mail urgences).

A scraping experiment in the Polish media, such as TVP, Rzeczpospolita, Gazeta Wyborcza, Fakt, etc., is therefore being prepared to obtain an appropriate single-sentence dataset for publication – it is also going to be another incremental step toward the dissertation on the overall topic of NLP fact-checking and its stages, focusing on English and West Slavic languages.

### 4.3.3 Crowd-sourced fact-checking platform

In 2023, other members of our team [Bútora, 2023] with funding from Avast developed a crowd-sourced fact-checking platform[4], where users gather reputations like on Wikipedia, by sharing check-worthy pieces of information found across the internet, and by their checking with sources.

While I am not directly involved in the implementation of the project apart from early consulting, experiments with FSV CUNI are to be launched, populating this platform with data and users. After the experiments, other data and applications will be delivered, and their processing will be another part of my dissertation project.

### 4.3.4 CTKFacts expansion

In 2021/2022, another round of the CTKFacts annotation experiment (see section 3.1.3) was carried out with the FSV CUNI students, yielding about 5K new data points, including, for example, claims extracted from the Czech Twitter.

The data is being cleaned and examined and will be attached to one of the other upcoming publications and presented in the dissertation thesis.

## 4.4 Pipeline modernization

As mentioned throughout the chapter 2, the state of the art in NLP has shifted dramatically over the last year, and another of the tasks I am currently working on is the modernization of our pipeline – Claim Generation, Information Retrieval, Natural Language Inference models – and appropriate use of LLMs in the tasks.

So far, I have successfully finetuned LLaMA-2 [Touvron et al., 2023b] for the claim generation task, and we have a LoRA finetuning experimental setup ready for the NLI models. The use of LLaMA-2 and its successors for our tasks will be a topic on its own, as most publicly available LLMs filter out the other languages and focus solely on English.

## 4.5 The grand scope

Overall, in brief points, the main topics of my dissertation are expected to be:

[4] https://factcheck.fel.cvut.cz

1. Introduction of the fact-checking task and its data, strong model baselines, and specific properties in the **West Slavic** context.

2. An integration of the step of **Claim generation** step into it, based on methods of abstractive summarization.

3. A delivery of reliable **metrics** for the tasks and their validation with expert-level humans.

4. Modernization of the automated fact-checking framework and solutions in English and Czech into the age of **Large Language Models**. Solutions were proposed already – based on proprietary black-box LLMs such as GPT-3.5 [Chen et al., 2023] – our next goal is to deploy open-source LLMs in-house, experiment with different architectures, fine-tuning tasks and data, improving the SOTA on our benchmark data.

   Due to our aim to produce transparent and reproducible research, using open-source LLMs is preferred over popular proprietary ones like GPT-4.

5. As the current instruction-tuned Large Language Models exhibit an **ability to explain their reasoning** [Saeed and Omlin, 2023], the methods of eXplainable AI (XAI) may also be integrated into our automated fact-checking framework, giving the fact-checker further insights what is behind the model classification.

6. Multiple **validation experiments** are planned with real-world fact-checkers[5] to testify to the usability of our solutions in the real world.

[5] Partners from CEDMO and other projects

Chapter 5

# AVeriTeC Paper

## 5.1 Introduction

We release a pipeline for fact-checking claims using evidence retrieved from the web consisting of two modules – a *retriever*, which picks the most relevant sources among the available knowledge store[1] and an *evidence & label generator* which generates evidence for the claim using these sources, as well as its veracity label.

Our pipeline is a variant of the popular Retrieval-augmented Generation (RAG) scheme [Lewis et al., 2020], making it easy to re-implement using established frameworks such as Langchain, Haystack, or our attached Python codebase for future research or to use as a baseline.

This paper describes our pipeline and the decisions taken at each module, achieving a simple yet efficient RAG scheme that improves dramatically across the board over the baseline system from [Schlichtkrull et al., 2024b], and scores third in the AVeriTeC leaderboard as of August 2024, with an AVeriTeC test set score of 50.4%.

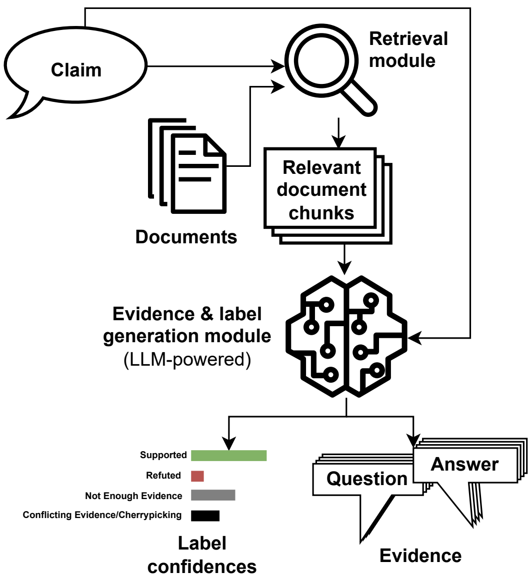


**Figure 5.1:** Our pipeline

[1]Due to the pre-retrieval step that was used to generate knowledge stores, our “retriever” module could more conventionally be referred to as a “reranker”, which we refrain from, to avoid confusion with reranking steps it uses as a subroutine.

## 5.2 Related work

1. **AVeriTeC shared task** [Schlichtkrull et al., 2024b] releases the datase of real-world fact-checked claims, annotated with evidence available at the date the claim was made.

   It proposes the **AVeriTeC Score** – a method of unsupervised scoring of fact-checking pipeline against this gold data using Hungarian METEOR score, matching the evidence questions (Q) or the whole evidence (Q+A). The score is then calculated as the proportion of claims with accurate label and sound evidence (using a threshold for Hu-METEOR such as 0.25) among all claims in the dataset, giving an estimate on "how often the whole fact-checking pipeline succeeds end to end".

   The provided **baseline** is a pipeline of search query generation, API search (producing a knowledge store), sentence retrieval, Question-and-answer (QA) generation, QA reranking, QA-wise claim classification and label aggregation, achieving an overall AVeriTeC test set score of 11%.

2. **FEVER Shared Task** [Thorne et al., 2018d], a predecessor to the AVeriTeC, worked with a similar dataset engineered on top of the enclosed domain Wikipedic data rather than real-world fact-checks. Its top-ranking solutions used a simpler pipeline of Document Retrieval, Sentence Reranking and Natural Language Inference, improving its modules in a decoupled manner and scoring well above 60% in a similarly computed FEVER score [Thorne et al., 2018b] on this data.

3. **Our previous research** on fact-checking pipelines [Ullrich et al., 2023b,Drchal et al., 2023] using data similar to FEVER and AVeriTeC shows significant superiority of fact-checking pipelines that **retrieve the whole documents** for the inference step, rather than retrieving out-of-context sentences.

4. **Retrieval-Augmented Generation (RAG) for Knowledge-Intensive Tasks** [Lewis et al., 2020] takes this a step further, leveraging Large Language Model (LLM) for the task, providing it the whole text of retrieved documents (each a chunk of Wikipedia) and simply instructing it to predict the evidence and label on top of it, achieving results within 4.3% from the FEVER state of the art by the time of its publication (December 2020) *without* engineering a custom pipeline for the task.

## 5.3 System description

Our system design prioritizes simplicity, and its core idea is using a straightforward RAG pipeline without engineering extra steps, customizing only the retrieval step and LLM prompting (Listing B.6 in Appendix B.2). Despite that, this section describes and justifies our decisions taken at each step, our additions to the naive version of RAG modules to tune them for the specific task of fact-checking, and their impact on the system performance.

### 5.3.1 Retrieval module

To ease comparison with the baseline and other systems designed for the task, our system does not use direct internet/search-engine access for its retrieval, but an AVeriTeC *knowledge store* provided alongside each claim.

To use our pipeline in the wild, our retrieval module is decoupled from the rest of the pipeline and can be swapped out in favour of an internet search module such as SerpApi[2] as a whole, or it can be used on top of a knowledge store emulated using large crawled corpora such as CommonCrawl[3] and a pre-retrieval module.

### Knowledge stores

Each claim's knowledge store contains pre-scraped results for various search queries that can be derived from the claim using human annotation or generative models. The knowledge stores used with ours as well as the baseline system can be downloaded from the AVeriTeC dataset page[4], containing about 1000 pre-scraped *documents*[5], each consisting of 28 sentences at median[6], albeit varying wildly between documents. The methods used for generating the knowledge stores are explained in more detail by [Schlichtkrull et al., 2024b].

Our retrieval module then focuses on picking a set of $k$ ($k = 10$ in the examples below, as well as in our submitted system) most appropriate document chunks to fact-check the provided claim within this knowledge store.

### Angle-optimized embedding search

Despite each article in any knowledge store only needing to be compared *once* with its *one specific* claim, which should be the use-case for CrossEncoder reranking [Déjean et al., 2024], our empirical preliminary experiments made us favour a *cosine-similarity* search based on vector embeddings instead. It takes less time to embed the whole knowledge store into vectors than to match each document against a claim using crossencoder, and the produced embeddings can be re-used across experiments.

For our proof of concept, we explore the MTEB [Muennighoff et al., 2023] benchmark leaderboard, looking for a reasonably-sized open-source embedding model, ultimately picking Mixedbread's mxbai-large-v1 [Li and Li, 2024, Lee et al., 2024] optimized for the cosine objective fitting our inteded use.

To reduce querying time at a reasonable exactness tradeoff, we use Faiss index [Douze et al., 2024, Johnson et al., 2019] to store our vectors, allowing us to only precompute semantical representation once, making the retriever respond rapidly in empirical experiments, allowing a very agile prototyping of novel methods to be used.

### Chunking with added context

Our initial experiments with the whole AVeriTeC documents for the Document Retrieval step have revealed a significant weakness – while most documents fit within the input size of the embedding model, outliers are common, often with *hundreds of thousands* characters, exceeding the 512 input tokens with little to no coverage of their content.

Upon further examination, these are typically PDF documents of legislature, documentation and communication transcription – highly relevant sources real fact-checker would scroll through to find the relevant part to refer.

[2] `https://serpapi.com/`

[3] `https://commoncrawl.org/`

[4] `https://fever.ai/dataset/averitec.html`

[5] The numbers are orientational and were computed on knowledge stores provided for the AVeriTeC dev set.

[6] devsetnote

This workflow inspires the use of *document chunk retrieval* as used in [Lewis et al., 2020], commonly paired with RAG. We partition each document into sets of its sentences of combined length of $N$ characters at most. To take advantage of the full input size of the vector embedding model we use for semantical search, we arbitrarily set our bound $N = 512 * 4 = 2048$, where 512 is the input dimension of common embedding models, 4 often being used as a rule-of-thumb number of characters per token for US English in modern tokenizers [OpenAI, 2023b].

Importantly, each chunk is assigned metadata – the source URL, as well as the full text of the next and previous chunk within the same document. This way, chunks can be presented to the LLM along with their original context in the generation module, where the length constraint is much less of an issue than in vector embedding. As shown in [Drchal et al., 2023], fact-checking models benefit from being exposed to larger pieces of text such as paragraphs or entire documents rather than out-of-context sentences. Splitting our data into the maximum chunks that fit our retrieval model and providing them with additional context may help down the line, preventing the RAG sources from being semantically incomplete.

### Pruning the chunks

While the chunking of long articles prevents their information from getting lost to retriever, it makes its search domain too large to embed on demand. As each of the thousands of claims has its own knowledge store, each of possibly tens of thousands of chunks, we seek to omit the chunks having little to no common tokens with our claim using an efficient BM25 [Robertson et al., 1995] search for the nearest $\omega$ chunks, setting the $\omega$ to 6000 for dev and 2000 for test claims. This yields a reasonably-sized document store for embedding each chunk into a vector, taking an average of 40 s to compute and store using the method described in Section 5.3.1 for each dev-claim using our Tesla V100 GPU.

This allows a quick and agile production of vectorstores for further querying and experimentation, motivated by the AVeriTeC test data being published just several days before the announced submission deadline. The pruning also keeps the resource intensity moderate for real-world applications. However, if time is not of the essence, the step can be omitted.

### Diversifying sources: MMR

Our choice of embedding search based on the entire claim rather than generating "search queries" introduces less noise and captures the semantics of the whole claim. It is, however, prone to redundancy among search results, which we address using a reranking by the results' Maximal Marginal Relevance (MMR) [Carbonell and Goldstein, 1998], a metric popular for the RAG task, which maximizes the search results' score computed as (for $D_i \in P$)

$$\lambda \cdot \mathrm{Sim}(D_i, Q) - (1 - \lambda) \cdot \max_{D_j \in S} \mathrm{Sim}(D_i, D_j)$$

$Sim$ denoting the cosine-similarity between embeddings, $Q$ being the search query, and $P$ the pre-fetched set of documents (by a search which simply maximizes their $Sim$ to $Q$), forming $S$ as the final search result, by adding each $D_i$ as MMR-argmax one by one, until reaching its desired size.

In our system, we set $\lambda = 0.75$ to favour relevancy rather than diversity, $|S| = 10$ and $|P| = 40$, obtaining a set of diverse sources relevant to each claim at a fraction of cost and complexity of a query-generation driven retrieval, such as that used in [Schlichtkrull et al., 2024b].

### 5.3.2 Evidence & label generator

The second and the last module on our proposed pipeline for automated fact checking is the Evidence & Label Generator, which receives a claim and $k$ sources (document chunks), and returns $l$ (in our case, $l = 10$) question-answer pairs of evidence abstracted from the sources, along with the veracity verdict – in AVeriTeC dataset, a claim may be classified as *Supported*, *Refuted*, *Not Enough Evidence*, or *Conflicting Evidence/Cherrypicking* with respect to its evidence.

Our approach leverages a Large Language Model (LLM), instructing it to output both evidence and the label in a single step, as a chain of thought. We rely on JSON-structured output generation with source referencing using a numeric identifier, we estimate the label confidences using Likert-scale ratings. The full system prompt can be examined in Listing B.6 in Appendix B.2, and this section further explains the choices behind it.

#### JSON generation

To be able to collect LLM's results programmatically, we exploit their capability to produce structured outputs, which is on the rise, with datasets available for tuning [Tang et al., 2024] and by the time of writing of this paper (August 2024), systems for strictly structured prediction are beginning to be launched by major providers [OpenAI, 2024].

Despite not having access to such structured-prediction API by the time of AVeriTeC shared task, the current generation of models examined for the task (section 5.3.2) rarely strays from the desired format if properly explained within a system prompt – we instruct our models to output a JSON of pre-defined properties (see prompt Listing B.6 in Appendix B.2) featuring both evidence and the veracity verdict for a given claims.

Although we implement fallbacks, less than 0.5% of our predictions threw a parsing exception throughout experimentation, and could be easily recovered using the same prompting again, exploiting the intrinsic randomness of LLM predictions.

#### Chain-of-thought prompting

While JSON dictionary should be order-invariant, we can actually exploit the order of outputs in our output structure to make LLMS like GPT-4o output better results [Wei et al., 2024]. This is commonly referred to as the "chain-of-thought" prompting – if we instruct the autoregressive LLM to first output the evidence (question, then answer), then a set of all labels with their confidence ratings (see section 5.3.2) and only then the final verdict, its prediction is both cheaper as opposed to implementing an extra module, as well as more reliable, as it must attend to all of the intermediate steps as well.

#### Source referring

To be able to backtrack the generated evidence to the urls of the used sources, we simply augment each question-answer pair with a source field. We assign a 1-based index[7] to each of the sources to facilitate tokenization and prompt the LLM to refer it as the source ID

[7] We chose the 1-based source indexing to exploit the source-referring data in LLM train set such as Wikipedia, where source numbers start with 1. The improvement in quality over 0-based indexing was not experimentally tested.

with each evidence it generates. While hallucination can not be fully prevented, it is less common than it may appear – with RAG gaining popularity, the models are being trained to cite their sources using special citation tokens [Menick et al., 2022], not dissimilarly to our proposal.

### Dynamic few-shot learning

To utilise the few-shot learning framework [Brown et al., 2020b] shown to increase quality of model output, we provide our LLMs with examples of what we expect the model to do. To obtain such examples, our evidence generator looks up the AVeriTeC train set using BM25 to get the 10 most similar claims, providing them as the few-shot examples, along their gold evidence and veracity verdicts. Experimentally, we also few-shot our models to output an *answer type* (*Extractive*, *Abstractive*, *Boolean*,...) as the *answer type* is listed with each sample anyways, and we have observed its integration into the generation task to slightly boost our model performance.

### Likert-scale label confidences

Despite modern LLMs being well capable of predicting the label in a "pick one" fashion, research applications such as ours may prefer them to output a probability distribution over all labels for two reasons.

Firstly, it measures the confidence in each label, pinpointing the edge-cases, secondly, it allows ensembling the LLM classification with any other model, such as Encoders with classification head finetuned on the task of Natural Language Inference (NLI) (see section 5.4.3).

As the LLMs and other token prediction schemes struggle with the prediction of continuous numbers which are notoriously hard to tokenize appropriately [Golkar et al., 2023], we come up with a simple alternative: instructing the model to print each of the 4 possible labels, along with their Likert-scale rating: 1 for "strongly disagree", 2 for "disagree", 3 for "neutral", 4 for "agree" and 5 for "strongly agree" [Likert, 1932].

On top of the ease of tokenization, Likert scale's popularity in psychology and other fields such as software testing [Joshi et al., 2015] adds another benefit – both the scale itself and its appropriate usage were likely demonstrated many times to LLMs during their unsupervised training phase.

To convert the ratings such as `{``Supported'':2, ``Refuted'':5, ``Cherrypicking'':4, ``NEE'':2}` to a probability distribution, we simply use softmax [Bridle, 1989]. While the label probabilities are only emulated (and may only take a limited, discrete set of values) and the system may produce ties, it gets the job done until further research is carried out.

### Choosing LLM

In our experiments, we have tested the full set of techniques introduced in this section, computing the text completion requests with:

1. GPT-4o (version `2024-05-13`)
2. Claude-3.5-Sonnet (`2024-06-20`), using the Google's Vertex API
3. LLaMA 3.1 70B, in the final experimets to see if the pipeline can be re-produced using open-source models

Their comparison can be seen in tables 5.1 and 5.2; for our submission in the AVeriTeC shared task, GPT-4o was used.

## 5.4 Other examined approaches

In this section, we also describe a third, optional module we call the *veracity classifier*, which takes the claim and its evidence generated by our evidence & label generator (section 5.3.2) and predicts the veracity label independently, based on the suggested evidence, using a fine-tuned NLI model. We also describe the options of its ensembling with veracity labels predicted in the generative step (section 5.3.2).

The absence of a dedicated veracity classifier has not been shown to decrease the performance of our pipeline significantly (as shown, e.g., in tables 5.2 and 5.1) so we suggest to omit this step altogether and we proceed to participate in the AVeriTeC shared task without it, proposing a clean and simple RAG pipeline without the extra step (Figure 5.1) for the fact-checking task.

### 5.4.1 Single-evidence classification with label aggregation

In the earliest stages of experimenting, we utilized the baseline classifier provided by AVeriTeC authors[8] [Schlichtkrull et al., 2024b]. It is based on the BERT [Devlin et al., 2019b] and was further fine-tuned on the AVeriTeC dataset [Schlichtkrull et al., 2024b]. It takes one claim and one question-answer evidence as input – each claim therefore has multiple classifications, one for each evidence. The classifications are then aggregated using a heuristic of several if-clauses to determine the final label.

We experiment with altering this heuristic (e.g. by making *not enough evidence* the final label only when no other labels are present at any evidence), and training NLI models that could work better with it, such as 3-way DeBERTaV3 [He et al., 2023] without a breakthrough result, motivating a radically different approach.

### 5.4.2 Multi-evidence classification

The multi-evidence approach is to fine-tune a 4-way Natural Language Inference (NLI) classifier, using the full scope of evidence directly at once, without heuristics. For that, we concatenate all of the evidence together using a separator `[SEP]` token. This allows the model to know exact question-answer borders, albeit using a space has turned out to be just as accurate as the experiments went on. As the veracity verdict should be independent of the evidence ordering, we also experiment with sampling different permutations in the fine-tuning step to increase the size of our data.

We carry out the fine-tuning using the AVeriTeC train split with gold evidence and labels on DeBERTaV3 [He et al., 2023] in two variants: the original large one[9] and one pre-finetuned on NLI tasks[10], and also Mistral-7B-v0.3 model[11] with a classification head (MistralForSequenceClassification) provided by the Huggingface Transformers library [Wolf et al., 2020] that utilizes the last token. In the preliminary testing phase, the original DeBERTaV3 Large performed the best and was used in all other experimental settings.

[8] `https://huggingface.co/chenxwh/AVeriTeC`
[9] `https://huggingface.co/microsoft/deberta-v3-large`
[10] `https://huggingface.co/cross-encoder/nli-deberta-v3-large`
[11] `https://huggingface.co/mistralai/Mistral-7B-v0.3`

From the approaches described above, we achieved the best results for the development split with gold evidence and labels with a model without permuting the evidence, achieving 0.71 macro $F_1$ score using a space-separation. The `[SEP]` model achieved a comparable 0.70 macro $F_1$ score, and the random order model performed worse with a 0.67 macro $F_1$ score, all improving significantly upon baseline, yet falling behind the capabilities of generating the labels alongside evidence in a single chain-of-thought. We provide our best DeBERTaV3 finetuned model publicly in a Huggingface repository[12].

### 5.4.3 Ensembling classifiers

Encouraged by the promising results of our multi-evidence classifiers, we go on to try to ensemble the models with LLM predictions from section 5.3.2, using a weighted average of the class probabilities of our models. We have experimented with multiple weight settings: 0.5:0.5 for even votes, 0.3:0.7 in favour of the LLM to exploit its accuracy while tipping its scales in cases of a more spread-out label probability distribution, as well as 0.1:0.9 to use the fine-tuned classifier only for tie-breaking, listing the results in Table 5.1.

We also tried tuning our ensemble weights based on a subset of the dev split, without a breakthrough in accuracy on the rest of dev samples.

The last method we tried was stacking using logistic regression. However, this setup classified no labels from *Not Enough Evidence* and *Conflicting Evidence/Cherrypicking*, and we could not achieve reasonable results. For logistic regression, we used the scikit-learn library [Pedregosa et al., 2011].

We conclude that the augmentation of the pipeline from Figure 5.1 with a classification module using a single NLI model or an ensemble with LLM is unneccessary, as it adds complexity and computational cost without paying off on the full pipeline performance (Table 5.2).

### 5.4.4 Conflicting Evidence/Cherrypicking detection

During the experiments, we discovered that classifying the *Conflicting Evidence/Cherrypicking* class is the most challenging task, achieving a near-zero $F_1$-score across our various prototype pipelines. To overcome this problem, we tried to build a binary classifier with cherrypicking as positive class. We tried to use the DeBERTaV3 Large model with both basic and weighted cross-entropy loss (other experimental settings were the same as in section 5.4.2), but it could not pick up the training task due to the *Conflicting Evidence/Cherrypicking* underrepresentation in train set – less than 7% of the samples carry the label.

Even after exploring various other methods, we did not get a reliable detection scheme for this task, perhaps motivating a future collection of data that represents the class better. While writing this system description paper, we found an interesting research by [Jaradat et al., 2024] that uses a radically different approach to detect cherrypicking in newspaper articles.

## 5.5 Results and analysis

We examine our pipeline results using two sets of metrics – firstly, we measure the prediction accuracy and $F_1$ over predict labels without any ablation, that is obtaining predicted labels using the predicted evidence generated on top the predicted retrieval results. While

[12] `https://huggingface.co/ctu-aic/deberta-v3-large-AVeriTeC-nli`

| Classifier | Acc | $F_1$ | Prec. | Recall |
|---|---|---|---|---|
| GPT4o | 0.72 | 0.46 | 0.48 | 0.47 |
| Claude 3.5 Sonnet | 0.64 | 0.49 | 0.50 | 0.52 |
| DeBERTa | 0.63 | 0.39 | 0.40 | 0.41 |
| DeBERTa - random@10 | 0.65 | 0.41 | 0.41 | 0.44 |
| $0.5 \cdot$ DeBERTa $+ 0.5 \cdot$ GPT4o | 0.70 | 0.43 | 0.41 | 0.45 |
| $0.5 \cdot$ DeBERTa $+ 0.5 \cdot$ Claude | 0.68 | 0.47 | 0.50 | 0.49 |
| $0.3 \cdot$ DeBERTa $+ 0.7 \cdot$ GPT4o | 0.72 | 0.45 | 0.45 | 0.46 |
| $0.3 \cdot$ DeBERTa $+ 0.7 \cdot$ Claude | 0.66 | **0.50** | **0.51** | **0.53** |
| $0.1 \cdot$ DeBERTa $+ 0.9 \cdot$ GPT4o | 0.72 | 0.39 | 0.46 | 0.43 |
| $0.1 \cdot$ DeBERTa $+ 0.9 \cdot$ Claude | 0.64 | 0.49 | 0.50 | 0.54 |
| Llama 3.1 | **0.73** | 0.44 | 0.43 | 0.46 |

**Table 5.1:** Evalution of the label generators, classifier models and their ensembles on the AVeriTeCdevelopment set. $F_1$, Precision and Recall are computed as macro-averages. The random@10 suffix indicates that the classifier used average of 10 different random orders of QA pairs for each claim. GPT4o stands for the Likert classifier based on GPT-4o, Claude 3.5 Sonnet is the Likert classifier based on Claude 3.5 Sonnet, and DeBERTa is classifier based on DeBERTaV3 Large fine-tuned on AVeriTeC gold evidence and labels.

| | Dev Set Scores | | | Test Set Scores | | |
|---|---|---|---|---|---|---|
| Pipeline Name | Q only | Q+A | AVeriTeC | Q only | Q+A | AVeriTeC |
| **GPT-4o (full-featured pipeline)** | **0.46** | **0.29** | **0.42** | 0.46 | **0.32** | **0.50** |
| GPT-4o (simplified) | 0.45 | 0.28 | 0.38 | 0.45 | 0.30 | 0.47 |
| Claude-3.5 (full-featured) | 0.43 | 0.28 | 0.35 | 0.42 | 0.30 | 0.46 |
| GPT-4o (with DeBERTa classifier) | 0.45 | 0.28 | 0.36 | – | – | – |
| AVeriTeC aseline | 0.24 | 0.19 | 0.09 | 0.24 | 0.20 | 0.11 |
| Llama 3.1 70B (full-featured) | **0.46** | 0.27 | 0.36 | **0.47** | 0.29 | 0.42 |

**Table 5.2:** Comparison of Pipeline Scores on Dev and Test Sets. Q, Q+A are Hu-METEOR scores against gold data, AVeriTeC scores are calculated as referred in section 5.2 thresholded at 0.25. "Full-featured" pipelines use the all the improvement techniques introduced in section 5.3, while the simplified pipeline omits the dynamic few-shot learning, answer-type-tuning and Likert-scale confidence emulation described in section 5.3.2

the retrieval module is fixed throughout the experiment (a full scheme described in section 5.3.1), various Evidence & Label generators and classifiers are compared in Table 5.1, showcasing their performance on the same sources. The results show that if we disregard the quality of evidence, models are more or less interchangeable, without a clear winner across the board – an ensemble of DeBERTA and Claude-3.5-Sonnet gives the best $F_1$ score, while GPT-4o scores 72% accuracy.

In real world, however, the evidence quality is critical for the fact-checking task. We therefore proceed to estimate it using the hu-METEOR evidence question score, QA score and AVeriTeC score benchmarks briefly explained in Section 5.2 and in greater detail in [Schlichtkrull et al., 2024b]. We use the provided AVeriTeC scoring script to calculate the values for Table 5.2, using its EvalAI blackbox to obtain the test scores without seeing the gold test data.

The latter experiments shown in Table 5.2 suggests the superiority of GPT-4o to predict the results for our pipeline with a margin. Even if we simplify the evidence & label generation step by omitting the dynamic few-shot learning (section 5.3.2), answer-type tuning and Likert-scale confidence emulation, it still scores above others, also showing that our pipeline can be further simplified when needed. Regardless of the LLM in use, the results of our pipeline improve upon the AVeriTeC baseline dramatically.

Posterior to the original experiments and to the AVeriTeC submission deadline, we also compute the pipeline results using an open-source model – the Llama 3.1 70B[13] [Dubey et al., 2024] obtaining encouraging scores, signifying our pipeline being adaptable to work well without the need to use a blackboxed proprietary LLM.

### 5.5.1 API costs

During our experimentation July 2024, we have made around 9000 requests to OpenAI's `gpt-4o-2024-05-13` batch API, at a total cost of \$363. This gives a mean cost estimate of \$0.04 per a single fact-check (or \$0.08 using the API without the batch discount) that can be further reduced using cheaper models, such as `gpt-4o-2024-08-06`.

We argue that such costs make our model suitable for further experiments alongside human fact-checkers, whose time spent reading through each source and proposing each evidence by themselves would certainly come at a higher price.

Our successive experiments with Llama 3.1 [Dubey et al., 2024] show promising results as well, nearly achieving parity with GPT. The use of open-source models such as LLaMa or Mistral allows running our pipeline on premise, without leaking data to a third party and billing anything else than the computational resources. For further experiments, we are looking to integrate them into the attached Python library using VLLM [Kwon et al., 2023].

### 5.5.2 Error analysis

In this section, we provide the results of an explorative analysis of 20 randomly selected samples from the development set. We divide our description of the analysis into the pipeline and dataset errors.

#### Pipeline errors

Our pipeline tends to rely on unofficial (often newspaper) sources rather than official government sources, e.g., with a domain ending or containing `gov`. On the other hand, it seems that the annotators prefer those sources. This could be remedied by implementing a different source selection strategy, preferring those official sources. For an example, see Listing B.1 in Appendix B.1.

Another thing that could be recognised as an error is that our pipeline usually generates all ten allowed questions (upper bound given by the task [Schlichtkrull et al., 2024b]). The analysis of the samples shows that the last questions are often unrelated or redundant to the claim and do not contribute directly to better veracity evaluation. However, since the classification step of our pipeline is not dependent on the number of question-answer pairs, this is not a critical error. Listing B.2 in Appendix B.1 shows an example of a data point with some unrelated questions.

When the pipeline generates extractive answers, it sometimes happens that the answer is not precisely extracted from the source text but slightly modified. An example of this error can be seen in Listing B.3 in Appendix B.1. This error is not critical, but it could be improved in future works, e.g. using post-processing via string matching.

Individual errors were also caused by the fact that we do not use the claim date in our pipeline and because our pipeline cannot analyse PDFs with tables properly. The

[13] `https://huggingface.co/hugging-quants/Meta-Llama-3.1-70B-Instruct-AWQ-INT4`

last erroneous behaviour we have noticed is that the majority of questions and answers are often generated from a single source. This should not be viewed as an error, but by introducing diversity into the sources, the pipeline would be more reliable when deployed in real-world scenarios.

#### Dataset errors

During the error analysis of our pipeline, we also found some errors in the AVeriTeC dataset that we would like to mention. In some cases, there is a leakage of PolitiFact or Factcheck.org fact-checking articles where the claim is already fact-checked. This leads to a situation where our pipeline gives a correct verdict using the leaked evidence. However, annotators gave a different label (often Not Enough Evidence). An example of this error is shown in Listing B.4 in Appendix B.1.

Another issue we have noticed is the inconsistency in the questions and answers given by annotators. Sometimes, they tend to be longer, including non-relevant information, while some are much shorter, as seen in Listing B.5 in Appendix B.1. The questions are often too general, or the annotators seem to use outside knowledge. This inconsistency in the dataset leads to a decreased performance of any models evaluated on this dataset.

#### Summary

Despite the abovementioned errors, the explorative analysis revealed that our pipeline consistently gives reasonable questions and answers for the claims. Most misclassified samples in those 20 data points were due to dataset errors.

## 5.6 Conclusion

In this paper, we describe the use and development of a RAG pipeline over real world claims and data scraped from the web for the AVeriTeC shared task. Its main advantage are its simplicity, consisting of just two decoupled modules – Retriever and an Evidence & Label Generator – and leveraging the trainable parameters of a LLM rather than on complex pipeline engineering. The LLMs capabilities may further improve in future, making the upgrades of our system trivial.

In section 5.3, we describe the process of adding features to both modules well in an iterative fashion, describing real problems we have encountered and the justifications of their solution, hoping to share our experience on how to make such systems robust and well-performing. We publish our failed approaches in section 5.4 and the metrics we observed to benchmark our systems in section 5.5. We release our Python codebase to facilitate further research and applications of our system, either as a baseline for future research, or for experimenting alongside human fact-checkers.

### 5.6.1 Future works

1. Integrating a search API for use in real-world applications
2. Re-examine the Likert-scale rating (section 5.3.2) to establish a more appropriate and fine-grained means of tokenizing the label probabilities
3. Generating evidence in the form of declarative sentences rather than Question-Answer pairs should be explored to see if it leads for better or worse fact-checking performance

4. RAG-tuned LLMs such as those introduced in [Menick et al., 2022] could be explored to see if they offer a more reliable source citing

# Chapter 6
# FEVER 8 Paper

## 6.1 Introduction

In 2024, Automated Verification of Textual Claims (AVeriTeC) shared task [Schlichtkrull et al., 2024a] showed that the fact checking of real-world claims like those from Politifact, AfricaCheck, etc., can be automated to a significant extent, with pipelines accessing Large Language Models (LLMs) to produce the evidence and veracity verdicts for previously unseen claims instead of a human. Almost each competitive AVeriTeC shared-task system, however, relied on a proprietary LLM like GPT-4o [Rothermel et al., 2024, Ullrich et al., 2024] or an open-weights model with high tens of billions of parameters [Yoon et al., 2024]. This raised a concern – can the fact-checking process be automated in a way accessible to masses, or is its quality conditioned by the business-owned blackbox models or access to prohibitive computational resources?

In this year's AVeriTeC shared task, the challenge is to match the quality of AVeriTeC systems with ones that only use open-weights models, constrained time of 60 seconds per claim on average, and a fixed compute of a single 23GB A10 GPU.

Our AIC CTU system (Figure 5.1), adapted for AVeriTeC from our last year submission, tops its test-leaderboard (Table 6.1) with a simple Retrieval-augmented Generation (RAG) scheme, using a locally hosted (Ollama) instance of Qwen3 LLM with 14B parameters, leveraging the sheer context length modern-day LLMs can process.

This paper introduces our system, discusses its design choices and how do they account on the score. We suggest our system as the new strong baseline – simple at core, competitive results – providing the code and reproduction advice.

## 6.2 System description

Our system is a straightforward adaptation of the AIC CTU Averitec system designed one year prior, published in [Ullrich et al., 2024]. The cited paper describes the system in detail, with ablation studies and justifications of each step. Our pipeline, depicted in Figure 6.1, consists of precomputation, retrieval, and generation modules:

i. Precomputation module

   1. The provided AVeriTeC **knowledge store** [Schlichtkrull et al., 2024b] is split into chunks of specified maximum length, each marked with metadata of its URL and the full texts of the chunk before and after.
   2. The chunks are then embedded into their vector representations, using only the chunk texts and no metadata.

3. Out of all chunk embeddings, a **vector store** is produced for each claim to be stored as a vector database.

ii. Retrieval module

1. The **Claim** is embedded into its vector representation using the same model used in i.2.
2. $k$ nearest neighbours are then retrieved from the vector store, along with their **chunk embeddings**
3. The chunk embeddings are then re-ranked using the Maximal Marginal Relevance (MMR) method [Carbonell and Goldstein, 1998], maximizing the embedding distances between retrieval results while minimizing their distance to the claim. Ultimately, we output a subset of $l$ diverse **sources** for the claim ($l < k$), augmenting each with its context before, after, and the text of its URL.

iii. Evidence & label generation module

1. We instruct a Large Language Model (LLM) to produce Question-Answer pairs required to fact-check given claim based on the provided sources, and predict its veracity verdict in a single output. We pass it the texts of all $l$ sources, and several few-shot QA-pair generation examples picked from Averitec train set retrieved using BM25 based on the tested claim. The whole instruction is serialized into a system prompt and the format we used can be seen in Appendix B.2.
2. **Claim** is then passed to the LLM as a user message.
3. LLM is called to **generate the evidence** as a Question-Answer-Source triples and the Likert-scale scores for each possible **veracity verdict** in a single prediction, performing a chain of thought.
4. The LLM output is parsed, and the verdict with the highest score is chosen for the claim.

The main differences between this year's AIC AVeriTeC system, opposed to last year's AIC AVeriTeC system, are the omission of knowledge store pruning in the precomputation step[1], and, importantly, the choice of LLM.

### 6.2.1 Model and parameter choices

To produce our submission in the AVeriTeC shared task, the following choices were made to deploy the pipeline from section 6.2:

`mxbai-embed-large-v1` [Li and Li, 2024, Lee et al., 2024] is used for the vector embeddings, and the maximum chunk size is set to 2048 characters, considering its input size of 512 tokens and a rule-of-thumb coefficient of 4 characters per token to exploit the full embedding input size and produce the smallest possible vector store size without neglecting a significant proportion of knowledge store text.

`FAISS` [Douze et al., 2024, Johnson et al., 2019] index is used as the vector database engine, due to its simplicity of usage, exact search feature and quick retrieval times (subsecond for a single AVeriTeC test claim).

$l = 10, k = 40, \lambda = 0.75$ are the parameters we use for the MMR reranking, meaning that 40 chunks are retrieved, 10 sources are yielded after MMR-diversification, and the

[1] The precomputed vector stores were required to be independent on claim text in AVeriTeC.

tradeoff between their similarity to the claim and their diversity is 3:1 in favour of the source similarity to the claim (explained in more detail in [Ullrich et al., 2024]).

`Ollama` wrapper around `llama.cpp` is the LLM engine we use to deploy LLMs within the AVeriTeC test environment due to its robustness and ease of deployment.

`Qwen3-14b` [Yang et al., 2025] is the LLM we use to produce the evidence and labels, we also let it generate its own <think> sequences, although further experimentation (Table 6.2) suggests that the thinking tokens may not justify the costs of their prediction, as they seem to perform on par with using only the evidence & label LLM outputs for its chain of thought.

## 6.3 Results and analysis

| **System** | ***old AVeriTeC score*** | ***Q only*** *($Ev^2R$)* | ***Q + A*** *($Ev^2R$)* | ***new AVeriTeC score*** | ***time per claim*** |
|---|---|---|---|---|---|
| AIC CTU | 0.41 | 0.20 | **0.48** | **0.33** | 54$s$ |
| HUMANE | 0.45 | 0.19 | 0.43 | 0.27 | 29$s$ |
| yellow flash | 0.16 | 0.16 | 0.41 | 0.25 | 32$s$ |
| FZIGOT | 0.46 | **0.36** | 0.40 | 0.24 | 19$s$ |
| EFC | 0.49 | 0.13 | 0.35 | 0.20 | **7$s$** |
| checkmate | 0.38 | 0.18 | 0.34 | 0.20 | 22$s$ |
| Baseline | **0.50** | 0.27 | 0.34 | 0.20 | 34$s$ |

**Table 6.1:** AVeriTeC shared task system leaderboard as shared by organizers, listing new $Ev^2R$-recall-based [Akhtar et al., 2024] and legacy hu-METEOR AVeriTeC scores. Evaluated using AVeriTeC 2025 test set. Best scores are bold.

In Table 6.1, we reprint the final test-leaderboard of AVeriTeC shared task as provided by the organizers. Our system introduced in Section 6.2 scores first in the decisive metric for the task – the new AVeriTeC score – with a significant margin. This came as a surprise to its authors, as neither the values of the old, hu-METEOR-based AVeriTeC score [Schlichtkrull et al., 2024b], nor the dev-leaderboard available during system development phase (where our system scored 4th), suggested its supremacy. Let us therefore proceed with a discussion of possible strengths that could have given our system an edge in verifying the AVeriTeC test-set of previously unseen 1000 claims.

### 6.3.1 Why does the system perform well?

So why should our system outperform the AVeriTeC baseline and even the other systems submitted to AVeriTeC shared task despite the simplicity of its design (Figure 6.1) which boils down to a straightforward case of retrieval-augmented generation (RAG)?

The main reason, in our experience, is the large **context size** we opt for – while even the AVeriTeC baseline processes the claims and sources in a manner more sophisticated than we do, it processes the knowledge store on a *sentence* level, reducing the amount of information passed to the LLM as opposed to working with *documents* as a whole, which is the strategy our system approximates.

Despite our proposed integration of LLM into the pipeline being rather vanilla, combining sources of total length of as much as 60K characters[2] on model input yields highly competitive results, leveraging its own trained mechanisms of context processing.

Our other advantages may have been using a very recent model, Qwen3 [Yang et al., 2025], which naturally has a slightly higher leakage of 2025 claims into its train set than older models, and outperforms the previous LLM generations at long sequence processing. Furthermore, our pipeline design only uses a single LLM call per claim, meaning we could use the generously-sized 14B variant of Qwen3 and still match the time limit with Nvidia A10 and 23GB VRAM.

### 6.3.2 Scoring change impact

While the new AVeriTeC score based on $Ev^2R$-recall [Akhtar et al., 2024] estimates the proportion of correctly fact-checked claims[3] in all claims, just like the old hu-METEOR-based AVeriTeC score did, their underlying methods differ. Most importantly, an LLM-as-a-judge approach is now used instead of symbolic evidence comparison method. The rise of our system from 3rd place in AVeriTeC shared task [Schlichtkrull et al., 2024a] to 1st place in AVeriTeC without any major system change[4] can therefore also be attributed to the used scoring method. The old scoring method was, for example, found to be prone to some level of noise, as it was not robust against evidence duplication [Malon, 2024], which was a found exploit to boost evidence recall.

The discrepancy between old and new AVeriTeC score in Table 6.1 could motivate a further study on how the new score behaves, for example using the test-prediction files from last year AVeriTeC shared task systems. The familiarity of the systems, the availability of their hu-METEOR scores and documentation, may reveal valuable insights into the $Ev^2R$ evaluation method itself, as in which behaviours does it punish and reward.

### 6.3.3 LLM impact

In 2024, we have experimented with then available versions of GPT-4o and Llama3.1-70B and found the open-source Llama to perform encouragingly well, depite the still-quite-cumbersome model size and the need for its quantization [Ullrich et al., 2024]. This year, we have simply gone for the most recent open-weights LLM at the largest parameter count we could fit within our AVeriTeC compute budget, thus choosing the Qwen3 at its 14B parameter size [Yang et al., 2025].

Qwen3 was trained to produce thinking tokens by default, an approach popularized by DeepSeek [DeepSeek-AI et al., 2025] and OpenAI research models, to force the chain of thought. We have experimented with enabling and disabling this feature to see if it has an impact on the AVeriTeC score, and compared the model output quality to our last year prediction dumps, with evaluation experiments listed in Table 6.2.

Both Qwen3 evidence and label generation settings perform on par with previous GPT-4o generation, which validates our model choice. The thinking tokens, while producing legitimate-looking writeups of the fact-checking workflows (see Appendix B.3) were not

[2]In other words, around 33 standard pages. This number follows from our parameter choices in Section 6.2.1: 10 sources are retrieved for each claim, each with $\sim 2048$ characters of the embedded text, and additional $\sim 4096$ characters of context.

[3]Claims with sound evidence w.r.t. human annotation, and an exact match in predicted label.

[4]Despite scaling down.

| LLM | Q only (Ev$^2$R) | Q + A (Ev$^2$R) | new AVeriTeC score |
|---|---|---|---|
| GPT-4o$_{\texttt{2024-05-13}}$ | 0.30 | 0.58 | 0.40 |
| Llama3.1-70B | 0.37 | 0.54 | 0.39 |
| qwen3:14B$_{\texttt{/no_think}}$ | 0.29 | 0.59 | 0.41 |
| qwen3:14B$_{\texttt{/think}}$ | 0.20 | 0.59 | 0.42 |

**Table 6.2:** Ablation study on LLM choice and <think>-tokens impact on AVeriTeC dev-score. Pipeline design (Figure 6.1), retrieval results, system and user prompts are fixed. Evaluated using an on-premise Ev$^2$R scorer with Ollama-hosted Llama3.3-70B as a judge.

shown to stimulate an improvement in AVeriTeC score in the ablation study (Table 6.2), so we suggest to disable this feature in future reproductions in favour of a faster prediction time (54s in the Table 6.1 was produced with the thinking feature *enabled*, so disabling it might solve the issue with near-limit runtime our pipeline suffers from).

## 6.4 Conclusion

In this paper, we have introduced our simple yet efficient RAG system which performed competitively well under time and compute constraints in AVeriTeC shared task, in May 2025. We release the used code along with usage instructions for producing the AVeriTeC submission, vector stores needed for the pipeline to run and their build scripts at `https://github.com/heruberuto/FEVER-8-Shared-Task/` which is a fork of the AVeriTeC baseline repository.

We attribute our success mostly to the use of *document* rather than *sentence* level of retrieval granularity and an employment of a recent LLM at a size which utilizes the whole compute and time budget with only around 10% time reserve as a failsafe. We encourage further usage of our system as a strong and easy-to-setup baseline for further research in automated fact checking and will be happy to answer any questions on the referred contacts.

### 6.4.1 Future works

1. Integrate a live search API as in [Malon, 2024] as a retriever into the AIC pipeline (Figure 6.1) to achieve a real-world generalization

2. Section 6.3.2 suggests to look at the key differences between legacy and Ev$^2$R scoring methods in terms of the available 2024 AVeriTeC leaderboard and available model documentations – we believe this could reveal valuable hints both scoring and pipelinne improvements in future work

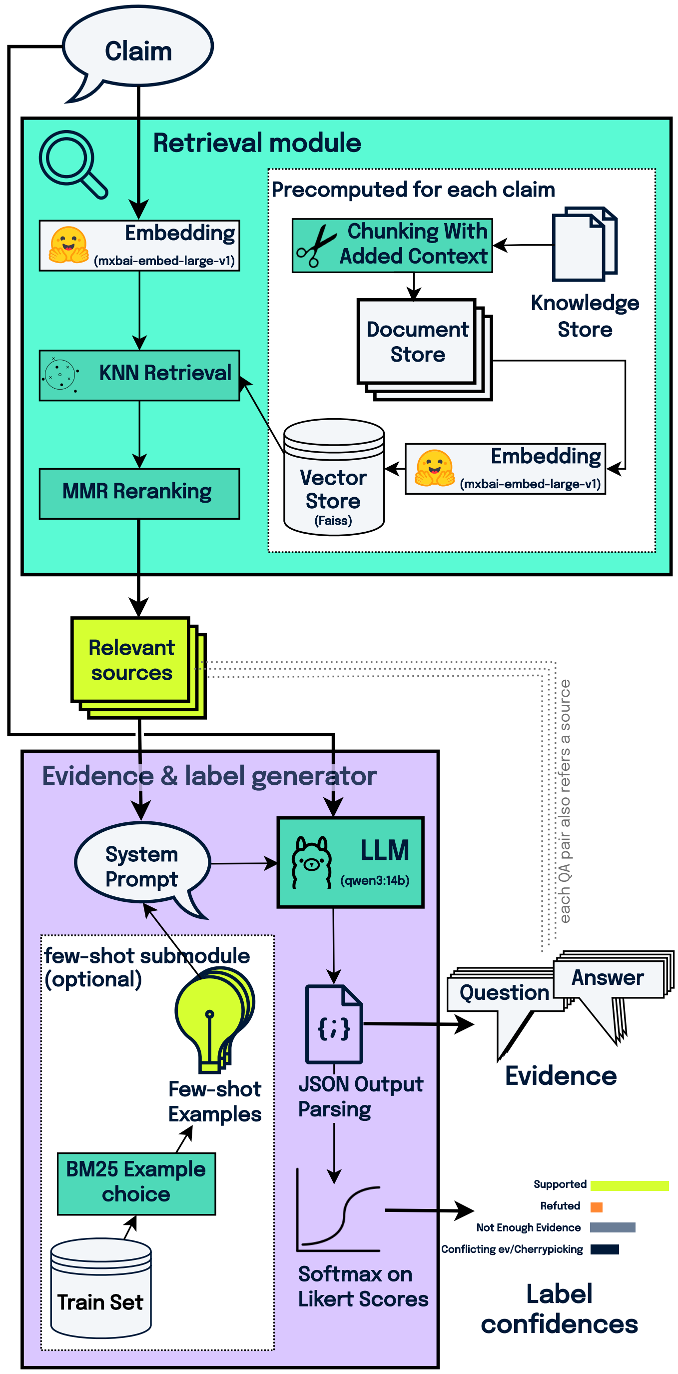


**Figure 6.1:** Our refreshed fact-checking pipeline used in CTU AIC FEVER 8 submission, adapted from [Ullrich et al., 2024].

# Chapter 7
# Conclusion

In this study, I have presented my current challenges and their motivation – a desire for an automated scheme to assist fact-checking. The solutions are being proposed in other literature and rely mostly on transformers, which is the current state of the art for nearly every NLP task. The transformer usage paradigm is shifting (from the approach of *fine-tuning* a *pre-trained* transformer to *prompting* or *few-shotting* a Large Language Model), which will impact my dissertation and also yield new challenges in modernizing our previous work.

So far, numerous datasets, most notably the CsFEVER and CTKFacts, have been collected, a working fact-checking pipeline was deployed on them, and the models we trained were published for further use.

Other tasks are to be established among the scientific public, importantly the claim generation and its model-based metrics, ongoing research such as the claim generation model training, collection of additional data in Czech, English, Polish, and Slovak is to be concluded, and new solutions for the whole problem of automated fact-checking are to be proposed, utilizing the new SOTA methods, such as the Large Language Models.

The point of the precedent chapters of the study was to give insights on what has been done so far, what is its value, what is the context in which this is happening, and what are the likely next steps in the future of my research.

# Bibliography


[Akhtar et al., 2024] Akhtar, M., Schlichtkrull, M., and Vlachos, A. (2024). Ev2r: Evaluating evidence retrieval in automated fact-checking.

[Allcott and Gentzkow, 2017] Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

[Aly et al., 2021] Aly, R., Guo, Z., Schlichtkrull, M. S., Thorne, J., Vlachos, A., Christodoulopoulos, C., Cocarascu, O., and Mittal, A. (2021). FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[Banerjee and Lavie, 2005] Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

[Barua et al., 2020] Barua, Z., Barua, S., Aktar, S., Kabir, N., and Li, M. (2020). Effects of misinformation on covid-19 individual responses and recommendations for resilience of disastrous consequences of misinformation. *Progress in Disaster Science*, 8:100119.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

[Bridle, 1989] Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In Touretzky, D., editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

[Brown et al., 2020a] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a). Language models are few-shot learners. *CoRR*, abs/2005.14165.

[Brown et al., 2020b] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020b). Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[Buchanan and Benson, 2019] Buchanan, T. and Benson, V. (2019). Spreading disinformation on facebook: Do trust in message source, risk propensity, or personality affect the organic reach of "fake news"? *Social Media + Society*, 5(4):2056305119888654.

[Bútora, 2023] Bútora, R. (2023). Crowd-sourcing platform frontend for fact-checking. `https://dspace.cvut.cz/handle/10467/109505`.

[Carbonell and Goldstein, 1998] Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

[Chen et al., 2017] Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051.

[Chen et al., 2023] Chen, J., Kim, G., Sriram, A., Durrett, G., and Choi, E. (2023). Complex claim verification with evidence retrieved in the wild.

[Cheng et al., 2016] Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733.

[Choi et al., 2021] Choi, E., Palomaki, J., Lamm, M., Kwiatkowski, T., Das, D., and Collins, M. (2021). Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

[Conneau et al., 2019] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

[Conneau et al., 2018] Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. (2018). Xnli: Evaluating cross-lingual sentence representations.

[DeepSeek-AI et al., 2025] DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., ..., and Zhang, Z. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

[Dettmers et al., 2023] Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms.

[Devlin et al., 2019a] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Devlin et al., 2019b] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Douze et al., 2024] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. (2024). The faiss library.

[Drchal et al., 2023] Drchal, J., Ullrich, H., Mlynář, T., and Moravec, V. (2023). Pipeline and dataset generation for automated fact-checking in almost any language.

[Dubey et al., 2024] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., ..., and Zhao, Z. (2024). The llama 3 herd of models.

[Déjean et al., 2024] Déjean, H., Clinchant, S., and Formal, T. (2024). A thorough comparison of cross-encoders and llms for reranking splade.

[Elsayed et al., 2021] Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Martino, G. D. S., and Atanasova, P. (2021). Overview of the clef-2019 checkthat!: Automatic identification and verification of claims.

[Fleiss, 1971] Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

[Fu et al., 2023] Fu, J., Ng, S.-K., Jiang, Z., and Liu, P. (2023). Gptscore: Evaluate as you desire.

[Gažo, 2021] Gažo, A. (2021). Algorithms for document retrieval in czech language supporting long inputs.

[Golkar et al., 2023] Golkar, S., Pettee, M., Eickenberg, M., Bietti, A., Cranmer, M., Krawezik, G., Lanusse, F., McCabe, M., Ohana, R., Parker, L., Blancard, B. R.-S., Tesileanu, T., Cho, K., and Ho, S. (2023). xval: A continuous number encoding for large language models.

[Guo et al., 2022] Guo, Z., Schlichtkrull, M., and Vlachos, A. (2022). A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

[Hanselowski et al., 2018] Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., and Gurevych, I. (2018). A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[Hasan et al., 2021] Hasan, T., Bhattacharjee, A., Islam, M. S., Samin, K., Li, Y., Kang, Y., Rahman, M. S., and Shahriyar, R. (2021). Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *CoRR*, abs/2106.13822.

[Hayes and Krippendorff, 2007] Hayes, A. and Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1:77–89.

[He et al., 2023] He, P., Gao, J., and Chen, W. (2023). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

[Holtzman et al., 2020] Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration.

[Hu et al., 2021] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

[Jaradat et al., 2024] Jaradat, I., Zhang, H., and Li, C. (2024). On context-aware detection of cherry-picking in news reporting.

[Ji et al., 2023] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

[Johnson et al., 2019] Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

[Joshi et al., 2015] Joshi, A., Kale, S., Chandel, S., and Pal, D. (2015). Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, 7:396–403.

[Kocián et al., 2021] Kocián, M., Náplava, J., Štancl, D., and Kadlec, V. (2021). Siamese bert-based model for web search relevance ranking evaluated on a new czech dataset.

[Koto et al., 2020] Koto, F., Baldwin, T., and Lau, J. H. (2020). Ffci: A framework for interpretable automatic evaluation of summarization.

[Krippendorff, 1970] Krippendorff, K. (1970). Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70.

[Krotil, 2022] Krotil, M. (2022). Text summarization methods in czech. `https://dspace.cvut.cz/handle/10467/101028`.

[Kwon et al., 2023] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

[Köpf et al., 2023] Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.-R., Stevens, K., Barhoum, A., Duc, N. M., Stanley, O., Nagyfi, R., ES, S., Suri, S., Glushkov, D., Dantuluri, A., Maguire, A., Schuhmann, C., Nguyen, H., and Mattick, A. (2023). Openassistant conversations – democratizing large language model alignment.

[Lazer et al., 2018] Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., and Zittrain, J. L. (2018). The science of fake news. *Science*, 359(6380):1094–1096.

[Lee et al., 2024] Lee, S., Shakir, A., Koenig, D., and Lipp, J. (2024). Open source strikes bread - new fluffy embeddings model.

[Lehečka and Švec, 2021] Lehečka, J. and Švec, J. (2021). Comparison of czech transformers on text classification tasks. In Espinosa-Anke, L., Martín-Vide, C., and Spasić, I., editors, *Statistical Language and Speech Processing*, pages 27–37, Cham. Springer International Publishing.

[Lewis et al., 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., K"u"ttler, H., Lewis, M., Yih, W.-t., Rockt"a"schel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[Li and Li, 2024] Li, X. and Li, J. (2024). AoE: Angle-optimized embeddings for semantic textual similarity. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand. Association for Computational Linguistics.

[Likert, 1932] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):55.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Liu et al., 2022a] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. (2022a). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

[Liu et al., 2023a] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

[Liu et al., 2020] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation.

[Liu et al., 2023b] Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., and Ge, B. (2023b). Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

[Liu et al., 2022b] Liu, Y., Liu, P., Radev, D., and Neubig, G. (2022b). BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

[Malon, 2024] Malon, C. (2024). Multi-hop evidence pursuit meets the web: Team papelo at FEVER 2024. In Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A., editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 27–36, Miami, Florida, USA. Association for Computational Linguistics.

[Menick et al., 2022] Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., and McAleese, N. (2022). Teaching language models to support answers with verified quotes.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Mlynář, 2023] Mlynář, T. (2023). Automated fact checking based on czech wikipedia. `https://dspace.cvut.cz/handle/10467/109219`.

[Mohri et al., 2023] Mohri, C., Andor, D., Choi, E., and Collins, M. (2023). Learning to reject with a fixed predictor: Application to decontextualization.

[Mroczkowski et al., 2021] Mroczkowski, R., Rybak, P., Wróblewska, A., and Gawlik, I. (2021). HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

[Muennighoff et al., 2023] Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive text embedding benchmark. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

[Nakov et al., 2021] Nakov, P., Martino, G. D. S., Elsayed, T., Barrón-Cedeño, A., Míguez, R., Shaar, S., Alam, F., Haouari, F., Hasanain, M., Mansour, W., Hamdan, B., Ali, Z. S., Babulkov, N., Nikolov, A., Shahi, G. K., Struß, J. M., Mandl, T., Kutlu, M., and Kartal, Y. S. (2021). Overview of the clef–2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news.

[Narayan et al., 2018] Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

[Nie et al., 2019a] Nie, Y., Chen, H., and Bansal, M. (2019a). Combining fact extraction and verification with neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

[Nie et al., 2019b] Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2019b). Adversarial NLI: A new benchmark for natural language understanding. *CoRR*, abs/1910.14599.

[Nørregaard and Derczynski, 2021] Nørregaard, J. and Derczynski, L. (2021). DanFEVER: claim verification dataset for Danish. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 422–428, Reykjavik, Iceland (Online). Link"o"ping University Electronic Press, Sweden.

[OpenAI, 2023a] OpenAI (2023a). Gpt-4 technical report.

[OpenAI, 2023b] OpenAI (2023b). What are tokens and how to count them? Accessed: 15 August 2024.

[OpenAI, 2024] OpenAI (2024). Introducing structured outputs in the api. Accessed: 15 August 2024.

[Pan et al., 2021] Pan, L., Chen, W., Xiong, W., Kan, M.-Y., and Wang, W. Y. (2021). Zero-shot fact verification by claim generation.

[Patel and Ahmad, 2023] Patel, D. and Ahmad, A. (2023). Google "we have no moat, and neither does openai". `https://www.semianalysis.com/p/google-we-have-no-moat-and-neither`. Accessed: 2023-09-06.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[Pikuliak et al., 2021] Pikuliak, M., Štefan Grivalský, Konôpka, M., Blšták, M., Tamajka, M., Bachratý, V., Šimko, M., Balážik, P., Trnka, M., and Uhlárik, F. (2021). Slovakbert: Slovak masked language model.

[Pomerlau and Rao, 2017] Pomerlau, D. and Rao, D. (2017). Fake news challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. `http://www.fakenewschallenge.org`. Accessed: 2023-09-06.

[Popel et al., 2020] Popel, M., Tomkova, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar, O., and Žabokrtský, Z. (2020). Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

[Raffel et al., 2019] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

[Robertson et al., 1995] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*.

[Rothermel et al., 2024] Rothermel, M., Braun, T., Rohrbach, M., and Rohrbach, A. (2024). InFact: A strong baseline for automated fact-checking. In Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A., editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112, Miami, Florida, USA. Association for Computational Linguistics.

[Rýpar, 2021] Rýpar, M. (2021). Methods of document retrieval for fact-checking. https://www.overleaf.com/read/thbvcjvvvfjp. [Online; accessed 21-May-2021].

[Saeed and Omlin, 2023] Saeed, W. and Omlin, C. (2023). Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273.

[Schlichtkrull et al., 2024a] Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A. (2024a). The automated verification of textual claims (AVeriTeC) shared task. In Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A., editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

[Schlichtkrull et al., 2024b] Schlichtkrull, M., Guo, Z., and Vlachos, A. (2024b). Averitec: a dataset for real-world claim verification with evidence from the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

[Schuster et al., 2021] Schuster, T., Fisch, A., and Barzilay, R. (2021). Get your vitamin c! robust fact verification with contrastive evidence. *CoRR*, abs/2103.08541.

[Sebastian, 2023] Sebastian, G. (2023). Exploring ethical implications of chatgpt and other ai chatbots and regulation of disinformation propagation.

[Semin, 2023] Semin, D. (2023). Multitask learning for nlp classifiers. https://dspace.cvut.cz/handle/10467/109243.

[Sido et al., 2021] Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., and Konopík, M. (2021). Czert – czech bert-like model for language representation.

[Štefánik et al., 2023] Štefánik, M., Kadlčík, M., Gramacki, P., and Sojka, P. (2023). Resources and few-shot learners for in-context learning in Slavic languages. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 94–105, Dubrovnik, Croatia. Association for Computational Linguistics.

[STEM, 2021] STEM (2021). Mýtům a konspiracím o covid-19 věří více než třetina české internetové populace | stem.cz. https://www.stem.cz/mytum-a-konspiracim-o-covid-19-veri-vice-nez-tretina-ceske-internetove-populace/. Accessed: 2021-05-03.

[Straka et al., 2018] Straka, M., Mediankin, N., Kocmi, T., Žabokrtský, Z., Hudeček, V., and Hajič, J. (2018). SumeCzech: Large Czech news-based summarization dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[Straka et al., 2021] Straka, M., Náplava, J., Straková, J., and Samuel, D. (2021). RobeCzech: Czech RoBERTa, a monolingual contextualized language representation model. *Lecture Notes in Computer Science*, page 197–209.

[Stănescu, 2022] Stănescu, G. (2022). Ukraine conflict: the challenge of informational war. *SOCIAL SCIENCES AND EDUCATION RESEARCH REVIEW*, 9(1):146–148.

[Szwoch et al., 2022] Szwoch, J., Staszkow, M., Rzepka, R., and Araki, K. (2022). Creation of Polish online news corpus for political polarization studies. In *Proceedings of the LREC 2022 workshop on Natural Language Processing for Political Sciences*, pages 86–90, Marseille, France. European Language Resources Association.

[Tang et al., 2024] Tang, X., Zong, Y., Phang, J., Zhao, Y., Zhou, W., Cohan, A., and Gerstein, M. (2024). Struc-bench: Are large language models really good at generating complex structured data?

[Taori et al., 2023] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Alpaca: A strong, replicable instruction-following model. `https://crfm.stanford.edu/2023/03/13/alpaca.html`. Accessed: 2023-09-04.

[DeepL, 2021] DeepL (2021). Deepl translator. `https://www.deepl.com/en/translator`. Accessed: 2021-05-09.

[Google, 2021] Google (2021). Cloud translation - google cloud. `https://cloud.google.com/translate`. Accessed: 2021-05-09.

[NLP-Progress, 2023] NLP-Progress (2023). On summarization. `http://nlpprogress.com/english/summarization.html`. Accessed: 2023-09-06.

[Thorne et al., 2018a] Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018a). FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

[Thorne et al., 2018b] Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018b). FEVER: a large-scale dataset for fact extraction and VERification. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

[Thorne et al., 2018c] Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A. (2018c). The Fact Extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

[Thorne et al., 2018d] Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A. (2018d). The fact extraction and VERification (FEVER) shared task. In Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A., editors, *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

[Thorne et al., 2019] Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A. (2019). The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

[Touvron et al., 2023a] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023a). Llama: Open and efficient foundation language models.

[Touvron et al., 2023b] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023b). Llama 2: Open foundation and fine-tuned chat models.

[Ullrich, 2021] Ullrich, H. (2021). Dataset for automated fact checking in czech language. `https://dspace.cvut.cz/handle/10467/95430`.

[Ullrich et al., 2023a] Ullrich, H., Drchal, J., Rýpar, M., Vincourová, H., and Moravec, V. (2023a). Csfever and ctkfacts: acquiring czech data for fact verification. *Language Resources and Evaluation*.

[Ullrich et al., 2023b] Ullrich, H., Drchal, J., Rýpar, M., Vincourová, H., and Moravec, V. (2023b). Csfever and ctkfacts: acquiring czech data for fact verification. *Language Resources and Evaluation*, 57(4):1571–1605.

[Ullrich et al., 2024] Ullrich, H., Mlynář, T., and Drchal, J. (2024). AIC CTU system at AVeriTeC: Re-framing automated fact-checking as a simple RAG task. In Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A., editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 137–150, Miami, Florida, USA. Association for Computational Linguistics.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Vicuna, 2023] Vicuna (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. `https://vicuna.lmsys.org/`. Accessed: 2023-09-04.

[Wang et al., 2020] Wang, A., Cho, K., and Lewis, M. (2020). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

[Wardle and Derakhshan, 2017] Wardle, C. and Derakhshan, H. (2017). *INFORMATION DISORDER : Toward an interdisciplinary framework for research and policy making Information Disorder Toward an interdisciplinary framework for research and policy-making.*

[Wei et al., 2024] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2024). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

[Williams et al., 2018] Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

[Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

[Wright et al., 2022] Wright, D., Wadden, D., Lo, K., Kuehl, B., Cohan, A., Augenstein, I., and Wang, L. L. (2022). Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland. Association for Computational Linguistics.

[Yamada et al., 2020] Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. (2020). Luke: Deep contextualized entity representations with entity-aware self-attention.

[Yang et al., 2025] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Zhou, F. H. . Z., and Qiu, Z. (2025). Qwen3 technical report.

[Yasunaga et al., 2021] Yasunaga, M., Leskovec, J., and Liang, P. (2021). LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Yoon et al., 2024] Yoon, Y., Jung, J., Yoon, S., and Park, K. (2024). HerO at AVeriTeC: The herd of open large language models for verifying real-world claims. In Schlichtkrull, M., Chen, Y., Whitehouse, C., Deng, Z., Akhtar, M., Aly, R., Guo, Z., Christodoulopoulos, C., Cocarascu, O., Mittal, A., Thorne, J., and Vlachos, A., editors, *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 130–136, Miami, Florida, USA. Association for Computational Linguistics.

[Zha et al., 2023] Zha, Y., Yang, Y., Li, R., and Hu, Z. (2023). Alignscore: Evaluating factual consistency with a unified alignment function.

[Zhang et al., 2020] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

[Zhang* et al., 2020] Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

[Zhao et al., 2023] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models.

[Šuppa and Adamec, 2020] Šuppa, M. and Adamec, J. (2020). A summarization dataset of Slovak news articles. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6725–6730, Marseille, France. European Language Resources Association.

# Appendix A
# Acronyms

**BERT** Bidirectional Encoder Representations from Transformers

**GPT** Generative Pre-trained Transformer

**FEVER** Fact Extraction and Verification – series of Shared tasks focused on fact-checking

**IR** Information Retrieval

**SOTA** State of the Art

**XSum** Extreme Summarization – summarizing article into one sentence

**NLI** Natural Language Inference

**ČTK** Czech Press Agency

# Appendix B
# Prompts and metaparameters

## B.1 Examples of errors

```
Claim 479: Donald Trump said "When the anarchists started ripping down our statues and
  ↪ monuments, right outside, I signed an order immediately, 10 years in prison."

gold evidence example:
    question: What was the law signed by Trump regarding damaging federal property?
    answer: Trump signed an executive order that authorizes a penalty of up to 10 years
  ↪ in prison for damaging federal property. It does not say that it will automatically
  ↪  be a 10 year prison sentence.,
    source_url: https://web.archive.org/web/20210224033536/https://www.whitehouse.gov/
  ↪ presidential-actions/executive-order-protecting-american-monuments-memorials-
  ↪ statues-combating-recent-criminal-violence/

pipeline evidence example:
    question: Did Trump sign an order related to vandalism of statues and monuments?,
    answer: Yes, Trump signed an executive order to prosecute those who damage national
  ↪ monuments, making it a punishable offense with up to 10 years in jail.,
    url: https://m.economictimes.com/news/international/world-news/trump-makes-
  ↪ vandalising-national-monuments-punishable-offence-with-up-to-10-yrs-jail/
  ↪ articleshow/76658610.cms
```

**Listing B.1:** Example of a claim where our pipeline uses newspaper sources instead of official government sources.

```
Claim 295: Trump campaign asked Joe Biden to release a list of potential Supreme Court
  ↪ picks only after Ginsburg's passing
question 1: Did Joe Biden claim that the Trump campaign asked him to release a list of
  ↪ potential Supreme Court picks only after Ginsburg's passing?
question 2: Did the Trump campaign ask Joe Biden to release a list of potential Supreme
  ↪ Court picks before Ginsburg's passing?
question 3: When did Trump release his latest list of potential Supreme Court nominees?
question 4: Did Trump personally demand that Biden release a list of potential Supreme
  ↪ Court nominees before Ginsburg's death?
question 5: What did Trump say about Biden releasing a list of potential Supreme Court
  ↪ nominees during the Republican National Convention?
question 6: Did the Trump campaign issue a statement on September 17, 2020, regarding
  ↪ Biden releasing a list of potential Supreme Court nominees?
question 7: What did the Trump campaign's statement on September 9, 2020, say about
  ↪ Biden releasing a list of potential Supreme Court nominees?
question 8: Did Biden indicate in June 2020 that he might release a list of potential
  ↪ Supreme Court picks?
quetion 9: What reason did Biden give for not releasing a list of potential Supreme
  ↪ Court nominees?,
question 10: Did Biden pledge to nominate a Black woman to the Supreme Court?
```

**Listing B.2:** Example of a claim and questions showing that the last tends to be unrelated or redundant to fact-checking of the claim.

```
Claim #155 - Trump said 'there were fine people on both side' in far-right protests.
answer: "You had some very bad people in that group, but you also had people that were
  ↪ very fine people, on both sides.",
answer_type: Extractive
url: https://www.theatlantic.com/politics/archive/2017/08/trump-defends-white-
  ↪ nationalist-protesters-some-very-fine-people-on-both-sides/537012/
scraped text: ... "You also had some very fine people on both sides," he said. The Unite
  ↪  the Right rally that sparked the violence in Charlottesville featured several
  ↪ leading names in the white-nationalist alt-right movement, and also attracted
  ↪ people displaying Nazi symbols. ...
```

**Listing B.3:** Example of a claim where our pipeline did not exactly extract the answer.

```
Claim #483 - Donald Trump said "We have spent nearly $2.5 trillion on completely
  ↪ rebuilding our military, which was very badly depleted when I took office."
Gold Label: Not Enough Evidence
Predicted Label: Refuted
pipeline evidence example:
    question: What is the total defense budget for the last four fiscal years under
  ↪ Trump?
    url: https://www.politifact.com/factchecks/2020/jan/10/donald-trump/trump-
  ↪ exaggerates-spending-us-military-rebuild/
    question: Did Trump spend $2.5 trillion specifically on rebuilding the military?
    url: https://www.factcheck.org/2020/07/trumps-false-military-equipment-claim/
    ...
```

**Listing B.4:** An example of a claim where the evidence consists mainly of evidence from PolitiFact and Factcheck.org fact-checking articles leading to different predicted label than in the gold dataset

```
Claim #0 - In a letter to Steve Jobs, Sean Connery refused to appear in an apple
  ↪ commercial.
Gold Evidence:
    question: Where was the claim first published
    answer: It was first published on Sccopertino
    question: What kind of website is Scoopertino
    answer: Scoopertino is an imaginary news organization devoted to ferreting out the
  ↪ most relevant stories in the world of Apple, whether or not they actually occurred -
  ↪  says their about page

Claim #315 - The fastest Supreme Court justice ever confirmed in the U.S. was 47 days.
Gold Evidence:
    question: What is the quickest time a Supreme Court justice nomination has been
  ↪ confirmed in the United States?
    answer: John Paul Stevens waited the fewest number of days (19)-followed by the most
  ↪  recent nominee to the Court, Amy Coney Barrett (27).61
    question: What is the average number of days between a nomination for a Supreme
  ↪ Court justice and the final Senate vote?
    answer: Overall, the average number of days from nomination to final Senate vote is
  ↪ 68.2 days (or approximately 2.2 months), while the median is 69.0 days.62 Of the 9
  ↪ Justices currently serving on the Court, the average number of days from nomination
  ↪  to final Senate vote is 72.1 days (or approximately 2.4 months), while the median
  ↪ is 73.0 days. Among the current Justices, Amy Coney Barrett waited the fewest
  ↪ number of days from nomination to confirmation (27), while Clarence Thomas waited
  ↪ the greatest number of days (99).
```

**Listing B.5:** An example of a claims which differs in length.

## B.2 System prompt

```
You are a professional fact checker, formulate up to 10 questions that cover all the facts needed to validate whether the factual statement (in User message) is true, false, uncertain or a matter of opinion. Each question has one of four answer types: Boolean, Extractive, Abstractive and Unanswerable using the provided sources.
After formulating Your questions and their answers using the provided sources, You evaluate the possible veracity verdicts (Supported claim, Refuted claim, Not enough evidence, or Conflicting evidence/Cherrypicking) given your claim and evidence on a Likert scale (1 - Strongly disagree, 2 - Disagree, 3 - Neutral, 4 - Agree, 5 - Strongly agree). Ultimately, you note the single likeliest veracity verdict according to your best knowledge.
The facts must be coming from these sources, please refer them using assigned IDs:
---
## Source ID: 1 [url]
[context before]
[page content]
[context after]
...

---
## Output formatting
Please, you MUST only print the output in the following output format:
```json
{
 "questions":
     [
         {"question": "<Your first question>", "answer": "<The answer to the Your first question>", "source": "<Single numeric source ID backing the answer for Your first question>", "answer_type":"<The type of first answer>"},
         {"question": "<Your second question>", "answer": "<The answer to the Your second question>", "source": "<Single numeric Source ID backing the answer for Your second question>", "answer_type":"<The type of second answer>"}
     ],
 "claim_veracity": {
     "Supported": "<Likert-scale rating of how much You agree with the 'Supported' veracity classification>",
     "Refuted": "<Likert-scale rating of how much You agree with the 'Refuted' veracity classification>",
     "Not Enough Evidence": "<Likert-scale rating of how much You agree with the 'Not Enough Evidence' veracity classification>",
     "Conflicting Evidence/Cherrypicking": "<Likert-scale rating of how much You agree with the 'Conflicting Evidence/Cherrypicking' veracity classification>"
 },
 "veracity_verdict": "<The suggested veracity classification for the claim>"
}
```
---
## Few-shot learning
You have access to the following few-shot learning examples for questions and answers.:

### Question examples for claim "{example["claim"]}" (verdict {example["gold_label"]})
"question": "{question}", "answer": "{answer}", "answer_type": "{answer_type}"
...
```

**Listing B.6:** System prompt for the LLMs, AVeriTeC claim is to be entered into the user prompt. Three dots represent omitted repeating parts of the prompt. Reprinted from [Ullrich et al., 2024].

## B.3 Qwen3 thinking example

```
<think>
Okay, let's tackle this query. The user is claiming that due to Imran Khan's criticism
  ↪ of Macron's comments on Islam, French authorities cancelled the visas of 183
  ↪ Pakistani citizens and deported 118 from the country. I need to verify this using
  ↪ the provided sources.

First, I'll look through the sources to see if there's any mention of visa
  ↪ cancellations or deportations related to Imran Khan's criticism.

Looking at Source ID 1, it talks about Imran Khan responding to Macron's remarks and
  ↪ the condemnation of Islamophobic policies, but there's no mention of visa
  ↪ cancellations or deportations.
...
Source ID 4 and 5 discuss diplomatic tensions, but not specific numbers of
  ↪ deportations. Source 5 and 6 from Newscomworld.com talk about fears of
  ↪ deportations of Pakistanis in France but don't mention specific numbers like 183
  ↪ or 118.
...
Source ID 10 mentions Pakistan summoning the French envoy but again, no numbers on
  ↪ deportations.

None of the sources mention 183 visas cancelled or 118 deported. The closest is Source
  ↪ 2, which mentions one individual, Muhammad Mansha, but that's a single case. The
  ↪ other sources discuss diplomatic tensions, protests, and possible fears of
  ↪ deportations but no actual numbers provided.

Therefore, the claim about 183 and 118 seems to be unsupported. The answer would be
  ↪ that there's no evidence of such numbers in the sources provided. The veracity
  ↪ verdict would be Refuted, as the sources don't support the specific numbers given.
</think>

```json
{
    "questions": [
        {"question": "Did French authorities cancel visas or deport Pakistani citizens
  ↪ in response to Imran Khan's criticism of Macron's comments on Islam?", "answer":
  ↪ "No specific information about visa cancellations or deportations of 183 Pakistani
  ↪ citizens or 118 deportees is mentioned in the provided sources.", "source": "10",
  ↪ "answer_type": "Boolean"},
        ...
        {"question": "What was the role of social media in the diplomatic dispute
  ↪ between Pakistan and France?", "answer": "Social media was used to spread calls
  ↪ for boycotts of French products and even nuclear attacks on France, but these were
  ↪ not confirmed by official sources.", "source": "5", "answer_type": "Extractive"}
    ],
    "claim_veracity": {
        "Supported": "1",
        "Refuted": "5",
        "Not Enough Evidence": "1",
        "Conflicting Evidence/Cherrypicking": "1"
    },
    "veracity_verdict": "Refuted"
}
```
```

**Listing B.7:** Qwen3 thinking tokens example.