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**
March 2026

Contents

Figures

Tables

Chapter 1
Introduction

TODO: přidat figuru, nezačínat rovnou mluvit o claimu, ale dát tomu nějaký primer

TODO: nějaký malý primer k claim extraction

TODO:

The dissertation you are holding in your hands, 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 (SOTA) 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 English, in the West Slavic languages (Czech, Slovak, and Polish) and multimodal setting where images and texts are combined. My contribution includes the collection and publication of novel datasets for the fact-checking task and its subroutines, trained models for these tasks, and metric frameworks that rate model behaviour in terms close to the human notion of *facticity* [Koto et al., 2020; Wright et al., 2022].

This dissertation covers 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 with source-grounded evidence. With the recent boom in NLP beginning with transformer networks and later Large Language Models (LLMs) [Zhao et al., 2023], few-shot learning [Brown et al., 2020] and prompting [Liu et al., 2023a], a significant part of this work is the adoption and evaluation of rapidly evolving SOTA NLP solutions in our specific context.

Overall, this dissertation follows up on my published research on fact-checking in Czech and extends it to other languages and modalities, while shifting from a pure *pre-training & fine-tuning* paradigm toward computationally feasible methods based on LLMs [Chen et al., 2023].

My central focus within the whole fact-checking scheme is the step of *claim generation*, treated as a benchmarkable NLP task adjacent to *abstractive summarization*. Benchmarking this task requires 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 dissertation presents the adopted directions, the reasoning behind them, the research questions, and the results obtained.

1.1 Research questions

The dissertation is structured around nine research questions introduced at the beginning of the relevant contribution chapters. Table 1.1 provides a compact overview and points to the chapter where each question is examined in detail.

	Research questions	
RQ1	To what extent do English automated fact-checking methods established by Fact Extraction and VERification (FEVER) shared tasks transfer to Czech, and what adaptations are necessary?	Chapter 3
RQ2	Can a Czech news-based fact-checking dataset be constructed as a usable benchmark by adapting the methodology established for English Wikipedia-based datasets?	Chapter 3
RQ3	Can a fact-checking baseline combining sparse retrieval and a Czech NLI model achieve meaningful performance on a newly constructed Czech news-domain benchmark?	Chapter 3
RQ4	Can the established Information Retrieval (IR) + Natural Language Inference (NLI) baseline for automated fact-checking be adapted to other languages using a fixed procedure and publicly available data to yield comparable results?	Chapter 4
RQ5	Can claim extraction be effectively modeled as abstractive summarization?	Chapter 5
RQ6	Can claim extraction quality be evaluated automatically in a way that aligns with human judgment?	Chapter 5
RQ7	Does a single LLM request per claim, with a prompt augmented by retrieval, yield competitive performance in automated fact-checking?	Chapter 6
RQ8	Does the RAG fact-checker still perform well when built on top of open-weights LLMs?	Chapter 6
RQ9	Can a text-based Retrieval-Augmented Generation (RAG) fact-checking pipeline be extended to image-text claims while remaining competitive?	Chapter 6

Table 1.1: Research questions addressed in this dissertation and the chapters where they are discussed.

1.2 Ideas & Roadmap

TODO: možná do conclusion?

TODO: This section will not be in the final print, it aims to identify a couple of findings that are not valuable enough to constitute a standalone paper, but would be strong and valuable as a section of the thesis. Currently, read more as an essay

1.2.1 Fact-checking limitations

1. **Targeting the unverifiable** oftentimes, the misinformation *intentionally* produces claims with no credible source of canonical information available to the reader –

such as intelligence from the war zones, the famous Iraqi WMD misinformation, etc. Typically, this information gets revealed with a significant delay, too late for the fact-check to influence the decision making or public opinion on the topic. *How serious is this?* Does it mean that the misinformer can always just obfuscate a bit more and the automated fact checking comes in vain? Are there impacted publications on this topic, credibility etc?

2. **Supportable misinformation** another area that interests me a lot are the on-line manipulations that are not technically false. Misinformer may use a couple of verifiable facts and present them in a misleading way. Podcaster Lex Fridman, for example, presents himself as a Ph.D. computer scientist coming from a background of teaching at MIT. While the information is technically not wrong, his affiliation is not what he makes it out to be – his degree was granted by the Drexel University, and as per the public record, his MIT teaching career comprised a single non-credit community class. This type of misinformation, in my view, reveals another blind spot of the automated fact-checking paradigm introduced in section 1.4 – while the abstraction of *evidence* and its retrieval may guarantee the neccessary explainability of automated fact-checking when the truths are broken, the *framing* of the truths seems to be just as important, with currently only classifier models being used to reveal manipulation techniques, with no rigorous explanation of their labels. There is actually literature and some whole *framing theory* on this, would be important to cite and draw practical conclusions from.

3. **Source reputation** was a topic of some Schlichtkrull Cambridge paper, and a round of research at our AIC NLP lab. Does it resolve the (1.) or does it bring about new problematic consequences? **TODO:** tidy up my thoughts on this **TODO:** Honza svoboda to dělá, obrátit se na něj a citnout mu závěrečnou práci

1.3 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].

Over the past years, 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] Hereinafter labeled as `NOT ENOUGH INFO`, in accordance to related research.

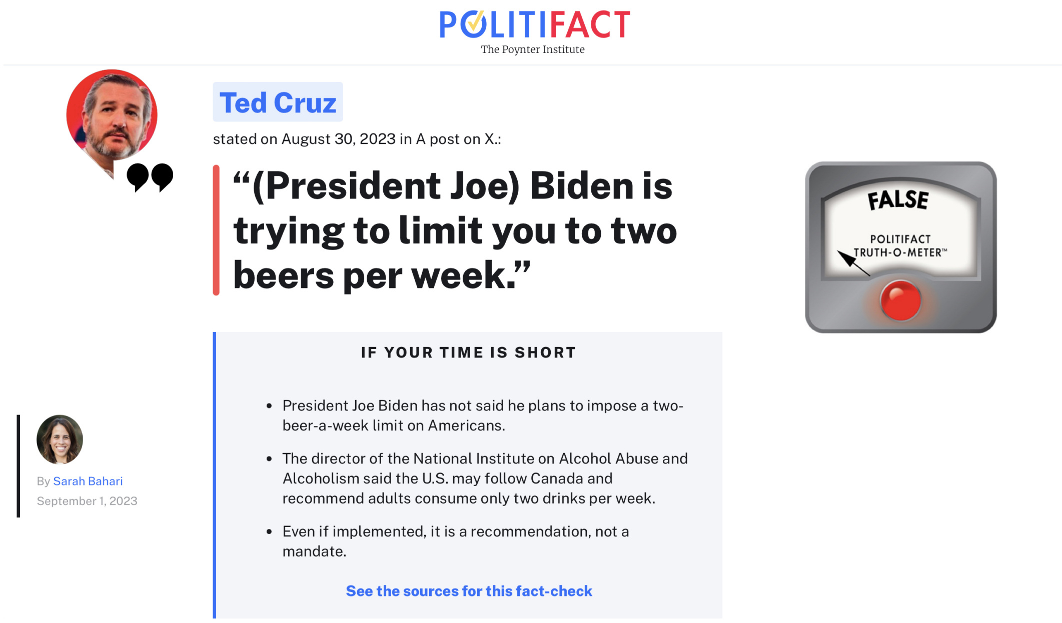

Figure 1.1: A real-world example of fact checking done by `https://politifact.org`

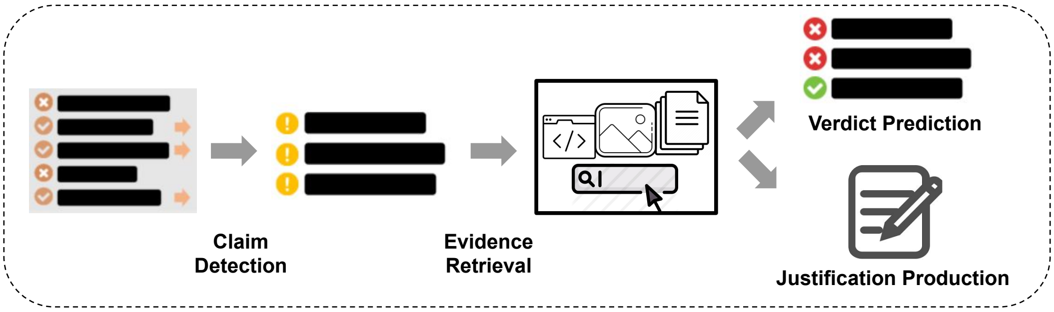

Figure 1.2: Automated fact-checking pipeline, reprinted from [Guo et al., 2022]

1.4 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, 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 FactCheck team at AIC CTU explores and adapts state-of-the-art methods for fact verification and related tasks in multiple languages, curates datasets, and develops strong baselines for Czech and English.

1.5 A word on the Transformers

For the past nine years, the SOTA solution for nearly every NLP 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 Bidirectional Encoder Representations from Transformers (BERT) encoder [Devlin et al., 2019] and its descendants, or the OpenAI's GPT-3 decoder [Brown et al., 2020] 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.

1.6 Obsoletion disclosure

Throughout this thesis, I will be, mostly chronologically, discussing the challenges and outcomes of my research spanning the last 5 years. This will include the experimental results and benchmarks obtained through experimentation with "then-SotA" methods in each task, to the rough date of the research question being my main focus.

Although this felt like the natural way of presenting my work in its full extent, it must be remarked that the field of my research, NLP, has been the fastest-developing field of knowledge in the past few years **TODO:** sources, with billions of dollars **TODO:** cite of funding coming to speed up its innovation. This led to obsoletion of my research in several ways: first, the models used in most experiments are dated and no longer represent the "best ability of current LLMs", which I often probed to see if a novel idea did or did not improve upon the general baseline. Second, proprietary models such as GPT-4 Turbo have come and gone during my doctoral studies, from their introduction as the new bleeding edge all the way to their removal from the API services that used to host them. The experiments with such models may therefore not be reproduced, and their substitution with a newer LLM may compromise the comparisons drawn from the original benchmarks. Third, entire research questions examined in the thesis may have lost a significant part of their relevance, now that a general-purpose LLM service with enabled internet search may perform reasonably reliable fact-checks tailored to custom user prompts **TODO:** figure grok is this true?.

In this thesis I aim not to obfuscate this fact, and while the following chapters introduce the original research questions, ideas, and the full experimentation from the time they came about, section 8.2 revisits the past works with the current (2026) generation of LLMs, discussing which parts of my work are now obsolete.

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

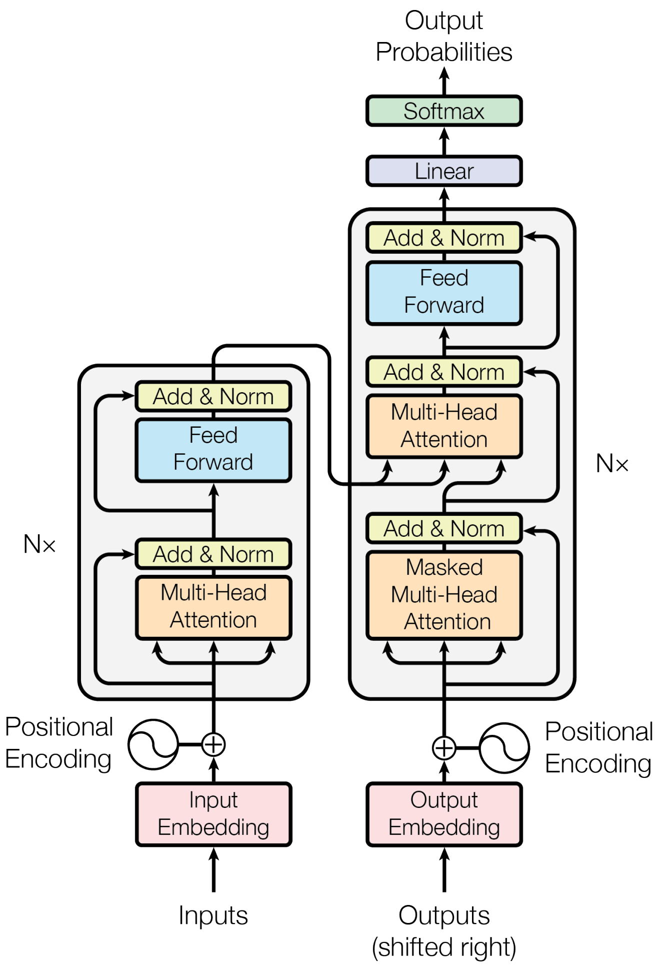

Figure 1.3: Transformer model architecture, reprinted from [Vaswani et al., 2017]

1.7 Dissertation outline

- **Chapter 1** introduces the dissertation topic, motivation, and scope
- **Chapter 2** examines the most relevant research and the shift from single-task fine-tuning to general-purpose LLMs
- **Chapter 3** explains our current contributions to the field of automated fact-checking and NLP in Czech
- **Chapter 4** presents the multilingual fact-checking pipeline and its language-transfer evaluation
- **Chapter 5** introduces the claim extraction task, dataset, metrics, and model comparisons
- **Chapter 6** describes our RAG-based fact-checking pipeline and shared-task submissions
- **Chapter 7** presents the agentic fact-checking direction and its system design

- Finally, **Chapter 8** concludes the dissertation with a wrap-up of findings

TODO: Synchronize this outline with the final chapter set if additional synthesis chapters are added.

Chapter 2
Related works

TODO: přejmenovat, případně rozdělit do více sota kapitol

This chapter reflects the SOTA in the field of Automated fact checking and NLP and the way it shifted during the years of my research. It describes the originally popular models for general NLP, such as BERT, and the recent paradigm shift from the *pretrain + finetune* transfer-learning framework popular since [Devlin et al., 2019] to currently dominant LLMs, which often outperform smaller models even without task-specific fine-tuning [OpenAI, 2023a; Touvron et al., 2023a; Vicuna, 2023]. It also covers optimization methods that enable training multi-billion-parameter pre-trained models on a single Graphics Processing Unit (GPU) and their relevance for this dissertation.

To connect these trends to our core topic, this chapter summarizes published approaches to automated fact-checking, 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 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 in 2018. BERT [Devlin et al., 2019] 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 SOTA results across various NLP benchmarks. Subsequently, numerous variations of pre-trained models like Generative Pre-trained Transformer (GPT) 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 [Lehecka and Svec, 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., 2020] 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., 2020].

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 LLMs like GPT-3 and subsequent iterations has spread the NLP hype even further. By utilizsing 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., 2020]. 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., 2022] 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 trend could shift SOTA NLP progress back toward open-source and public research, reducing any single-company "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., 2020], Github, Wikipedia, Gutenberg Project, Books3 corpus, ArXiv and Stack Exchange

Either way, these results show that open-source LLMs are a central approach for the NLP task of *Automated fact checking*.

2.4 Retrieval-augmented generation

Retrieval-augmented generation

2.5 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].

2.5.1 FEVER and followups

Soon, standard tasks began to be formulated and data collected. The Fact Extraction and VERification (FEVER) 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* – 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* (Recognizing Textual Entailment (RTE))

FEVER 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., 2018b, 2019], gives insights into the complexities of automated fact-checking task, and strong baselines for

[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*

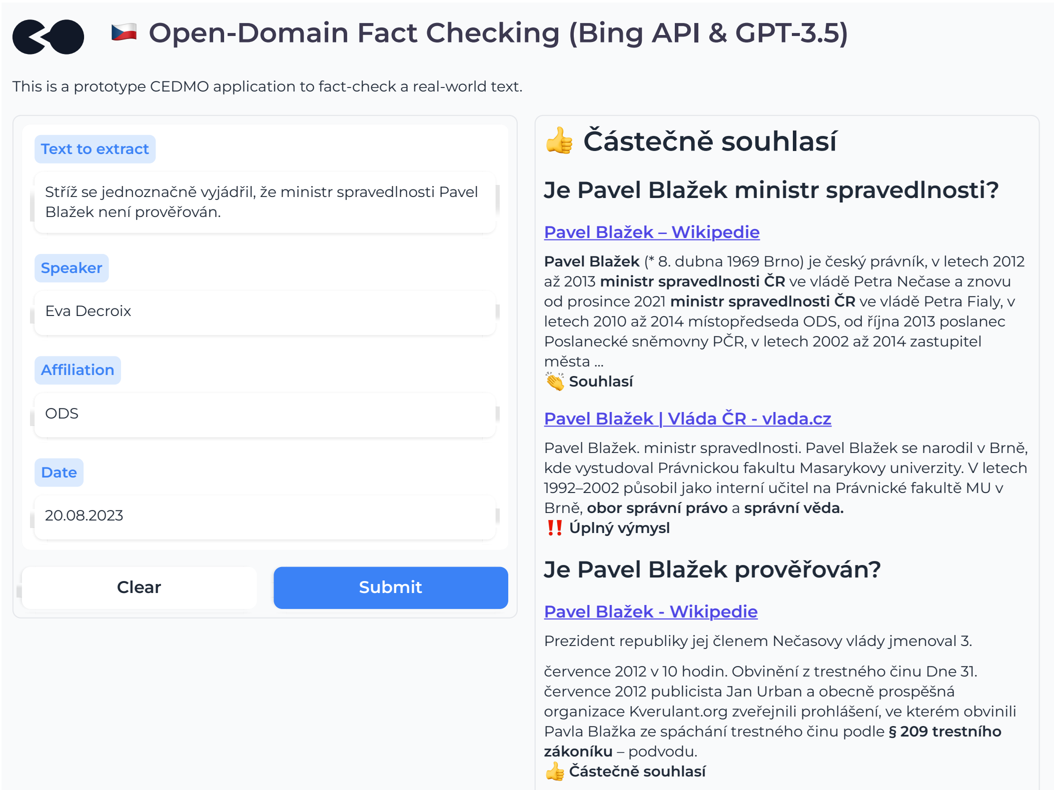

Figure 2.1: Proof-of-concept Czech fact-checking based on live-internet search (Bing API) and LLM prompting, based on the proposals of [Chen et al., 2023] in Czech, using a real-world claim that was fact-checked by `demagog.cz` in June 2023

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.

2.5.2 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

[5]See section 5.4.1

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).

From a contemporary perspective, this can be viewed as an early form of Retrieval-Augmented Generation (RAG) for fact-checking: external documents are retrieved first and then passed to an LLM to ground the final prediction.

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.

2.6 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 (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.

2.6.1 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.

These benchmarking limitations are addressed in Chapter 5, section 5.4, where the claim-extraction metric framework is formalized and validated.

Chapter 3
Current contribution

Research Questions

RQ1 To what extent do English automated fact-checking methods established by FEVER shared tasks transfer to Czech, and what adaptations are necessary?

RQ2 Can a Czech news-based fact-checking dataset be constructed as a usable benchmark by adapting the methodology established for English Wikipedia-based datasets?

RQ3 Can a fact-checking baseline combining sparse retrieval and a Czech NLI model achieve meaningful performance on a newly constructed Czech news-domain benchmark?

We collected novel data for fact-checking in our application context, emulated and scraped previously unavailable datasets, made these resources public where licensing allows it, established strong model baselines, and formalized 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., 2023] 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

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

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

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., 2023].

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., 2023] 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 data generation methods based on claim generation schemes such as [Pan et al., 2021]. **TODO:** Add a short thesis-level summary of which CsFEVER limitations are solved by the claim extraction pipeline in Chapter 5.

3.1.2 FCheck annotations platform

The imperfections in translated CsFEVER data, as well as collaboration with Czech Press Agency (ČTK) and the Faculty of Social Sciences, led me to design a natively Czech hand-annotated dataset that reduces translation noise and replaces rigid 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.

[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`

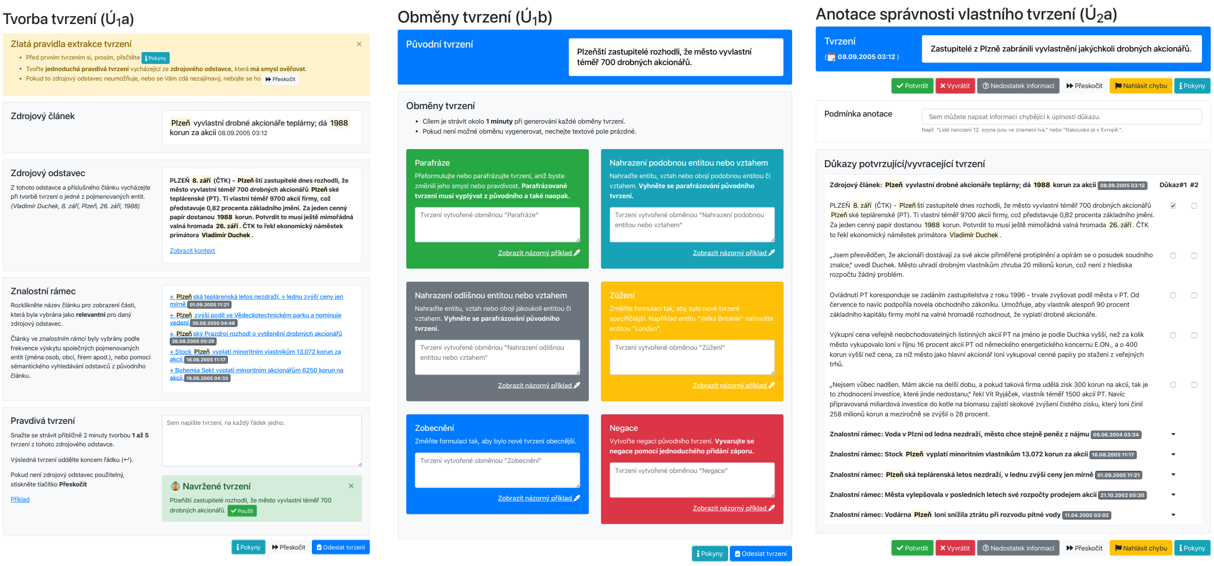

Figure 3.1: FCheck – a platform for fact-checking data collection developed for TAČR project; collects data for claim generation, information retrieval, and natural language inference tasks

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., 2023], 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., 2023]

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' κ-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' κ 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., 2023].

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 already being referred to and reused in external research [Stefanik et al., 2023].

3.1.4 Other NLP datasets in West Slavic languages

Over time, we have accumulated numerous datasets in Czech and other Slavic languages that had previously been poorly covered or unavailable. For the convenience of others,

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

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

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., 2020] 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 claim generation in Czech, we adapted related datasets and use:

   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`
   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`

Some of the data used to be restricted to private repositories, but with this dissertation most of them are published after licensing review. 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 – Natural Language Inference (NLI) 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., 2023] also focused on establishing a strong starting SOTA 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., 2019], achieving 75.9% macro-F1 and a significant superiority over previous shared task winner [Nie et al., 2019] (which had 69.5 macro-F1 with NSMNs)

3.2.2 Claim generation

In this dissertation, claim generation is treated as training models to output one or more factual claims as fluent, atomic, decontextualized, and faithful single sentences. In section 5.3, I propose the claim 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 2.6.1, 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 SOTA 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]

   We also trained variants 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.

The prototype models were iterated with our CEDMO[16] project partners (fact-checkers from European organizations), and their feedback informed model and interface refinements. **TODO:** Add a compact table with the final offline/partner evaluation outcomes used in this dissertation revision.

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 –

[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
[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

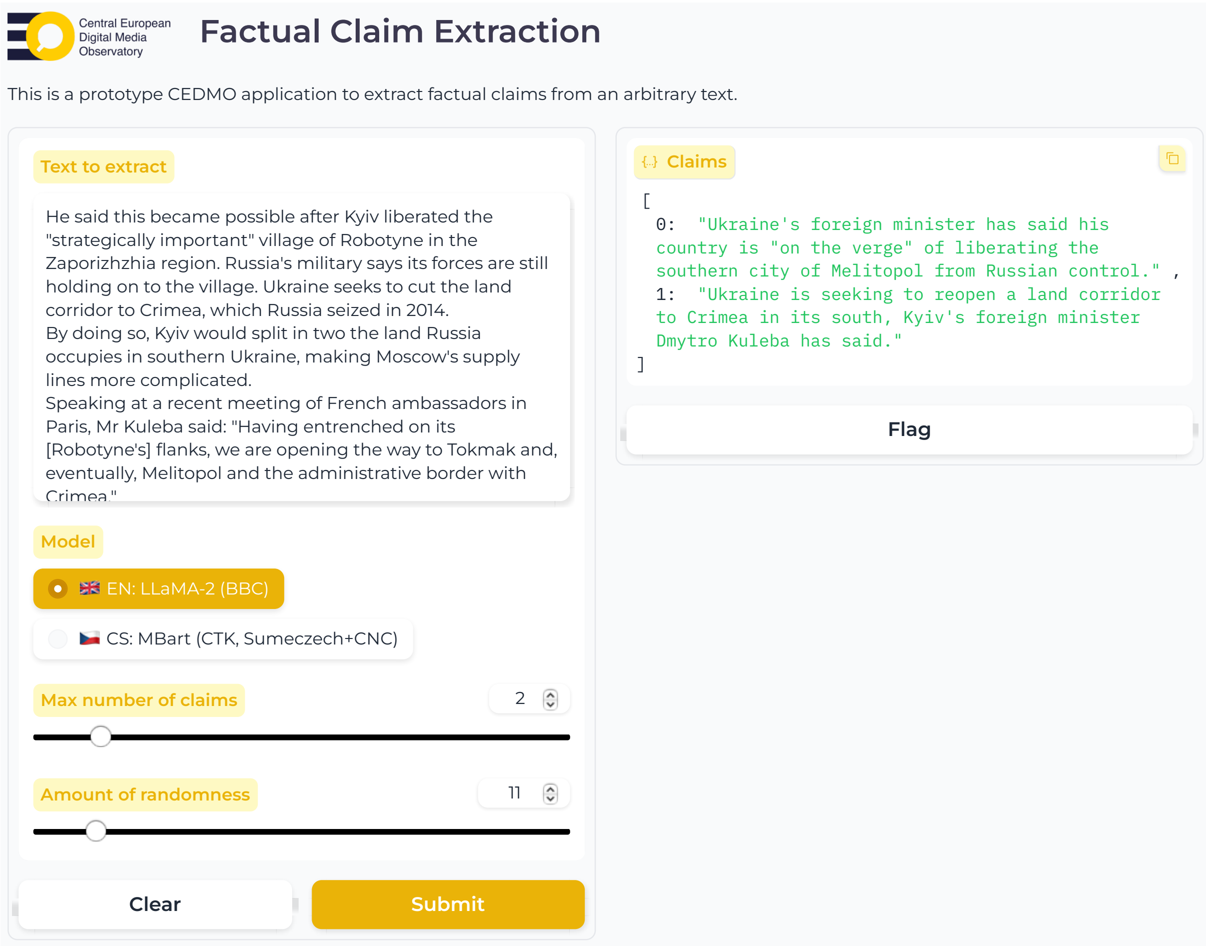

Figure 3.2: Factual claim extraction application done for the CEDMO project

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.

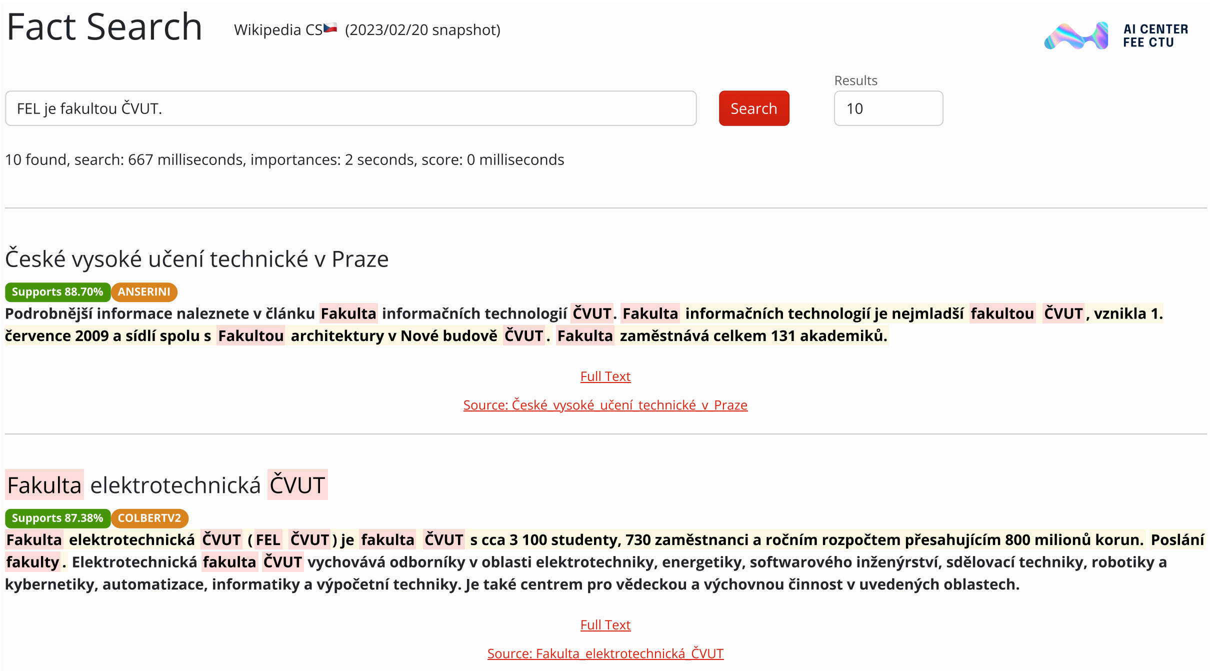

Figure 3.3: Automated fact-checking application "fact-search" verifying claims against Czech Wikipedia using our SOTA models

Chapter 4
Multilingual Fact-Checking Pipeline

Research Questions

RQ4 Can the established IR + NLI baseline for automated fact-checking be adapted to other languages using a fixed procedure and publicly available data to yield comparable results?

This chapter summarises the automated fact-checking pipeline published in [Drchal et al., 2024], a joint work with Jan Drchal, Tomáš Mlynář, and Václav Moravec. The pipeline targets the closed-world fact-checking paradigm – verifying claims against a fixed evidence corpus – and was designed with ease of language transfer as a central goal, demonstrated across Czech, English, Polish, and Slovak.

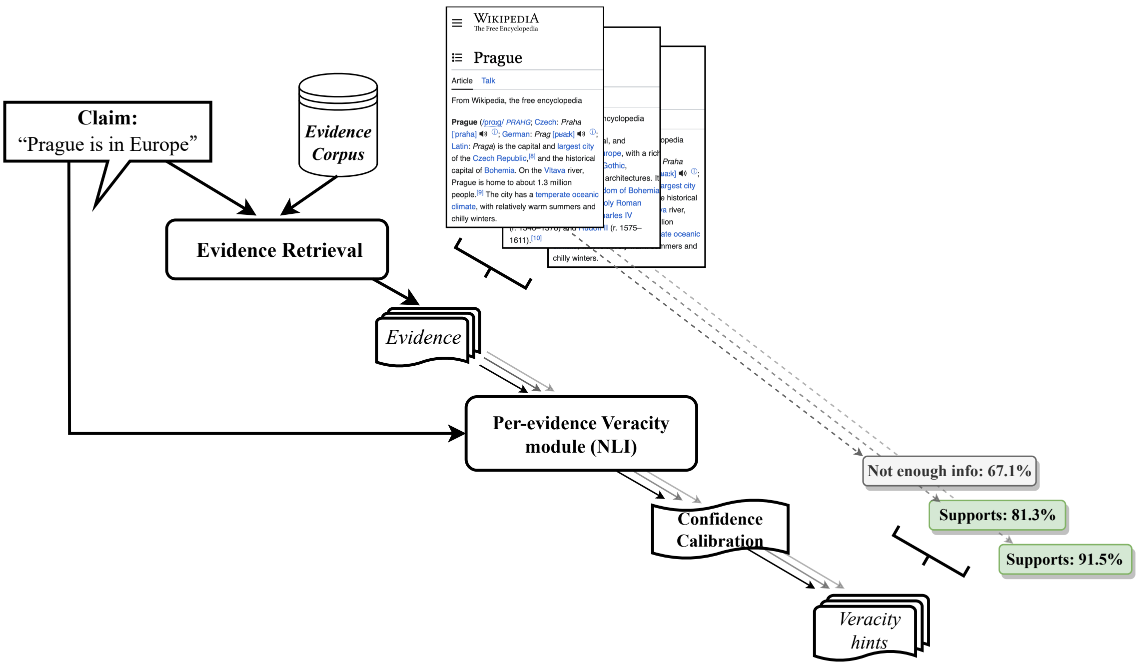

Figure 4.1: The automated fact-checking pipeline from [Drchal et al., 2024]. Given a claim, the *evidence retrieval* module fetches the most relevant paragraphs from a Wikipedia corpus. Each retrieved paragraph is then independently classified by the *veracity module*, yielding per-paragraph `SUPPORTS`/`REFUTES`/`NEI`labels rather than a single overall verdict.

4.1 Architecture

The pipeline (Figure 4.1) has two main modules. The **evidence retrieval** module retrieves the most relevant paragraphs from a Wikipedia snapshot for a given claim, using the ColBERTv2 late-interaction retrieval model. Working at the paragraph level – rather than the sentence level common in earlier FEVER-based pipelines [Thorne et al., 2018a] – removes the need for a separate sentence selection step and provides richer context to the downstream model. The **veracity module** classifies each retrieved paragraph independently against the claim using an XLM-RoBERTa-based NLI classifier [Conneau et al., 2020], calibrated via temperature scaling to produce reliable confidence estimates. Importantly, the pipeline deliberately reports *per-evidence* verdicts rather than a single black-box label, making the evidence visible to human fact-checkers and shifting the system towards an assistive rather than a fully automated role.

4.2 Data Generation

A key bottleneck for non-English pipelines is the absence of labelled training data. The pipeline addresses this through Question Answering for Claim Generation (QACG) [Pan et al., 2021], adapted for the paragraph level and for languages lacking the monolingual tooling required by the original work.

QACG operates in three steps given a source paragraph: (1) Named Entity Recognition (NER) extracts entities as candidate answers; (2) a question generation model produces a question whose answer is each entity; (3) a claim generation model converts each question–answer pair into a declarative claim. `SUPPORTS`claims are generated directly from the source paragraph; `REFUTES`claims substitute the answer entity with another entity of the same type sampled from the same paragraph; `NEI`claims are generated using entities from auxiliary paragraphs of the same Wikipedia article. Both the question generation and claim generation models require only moderate amounts of training data – SQuAD [Rajpurkar et al., 2016] and a question-to-declarative-sentence conversion dataset, respectively – which were machine-translated into Czech, Polish, and Slovak, keeping the translation overhead tractable. Full QACG datasets were generated from August 2023 Wikipedia snapshots for all four languages and released openly alongside all models and code.

4.3 Evaluation

The pipeline was evaluated quantitatively and via human annotation across all four languages. Results show that multilingual models trained on the concatenation of all four language datasets consistently outperform single-language models on both the evidence retrieval and NLI tasks. The limited domain transfer between QACG-generated and human-annotated FEVER data was analysed through Pointwise $\mathcal{V}$-information, revealing class-level difficulty differences attributable to the respective data generation strategies. Human annotations confirmed a claim failure rate of approximately 10% for QACG data – a known limitation from entity substitution errors – with limited impact on overall pipeline performance.

The pipeline was deployed as the *FactSearch* web application (Figure 3.3 in Chapter 3), whose initial user feedback confirmed the utility of the evidence-level presentation while highlighting room for improvement in NLI accuracy.

Full details, including all datasets, trained models, and source code, are available in [Drchal et al., 2024].

Chapter 5
Automated Claim Extraction

TODO: Rewrite chapter-specific prose to match dissertation voice and remove remaining paper-style copy where needed.

Research Questions

RQ5 Can claim extraction be effectively modeled as abstractive summarization?

RQ6 Can claim extraction quality be evaluated automatically in a way that aligns with human judgment?

5.1 Introduction

Recent research in Natural Language Processing has extensively covered the case of automated fact-checking as a pipeline of retrieval and inference tasks [Aly et al., 2022; Thorne and Vlachos, 2021]. As noted by Guo et al. 2022, these steps alone do not represent the whole challenge human fact-checkers face, notably omitting the step of coming up with the claims to be checked in the first place.

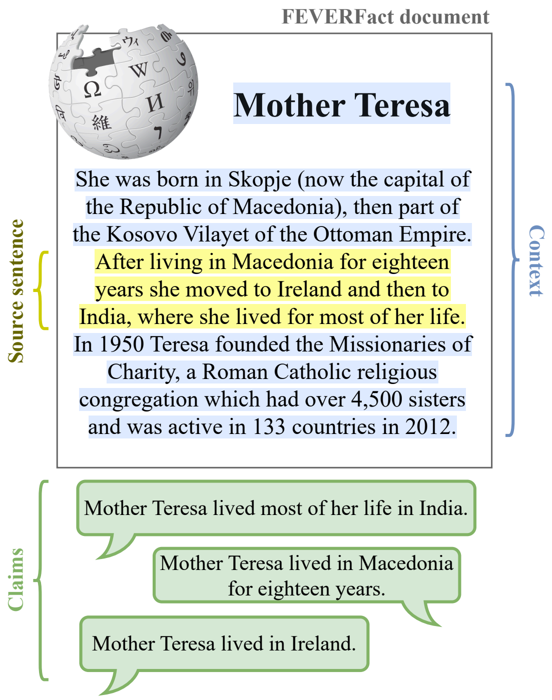

Figure 5.1: An example of claim extraction from the FEVERFACT dataset, constructed from data published by Thorne and Vlachos 2021. The three-sentence document (top) provides context; the middle sentence is the extraction target. The claims (bottom) are human-annotated atomic factual statements that can be extracted from it.

The real-world factual claims may be scattered throughout a politician's social network post, pieced together from a debate transcript, distilled from a lengthy news article, etc. Research such as Barrón-Cedeño et al. 2020 therefore proposes augmenting the fact-checking pipeline by the step of *claim detection*, commonly modelled as classification of *check-worthy* claims within a real-world text segmented into its sentences.

The claim detection paradigm, however, only kicks the can down the road – the detected "check-worthy" parts of text still need to be processed by a human fact-checker [Sheikhi et al., 2023] and re-written into the form of a *claim*: a factual statement presenting the source text information without the need of additional context in a single sentence.

The automated generation of such claims is desirable not only for the fact-checkers themselves [Deng et al., 2024], but also for further research in the field [Drchal et al., 2024] where the generation of claims on top of large-scale corpora was shown to yield valuable data to train models [Pan et al., 2021].

In this chapter, we explore the abstractive methods of claim extraction, modelled as one-to-many text-to-text task: after seeing the whole input text, the model is tasked to formulate a set of claims it makes, as simple self-contained sentences faithful to the source.[1] We construct a downstream task for the models – the **FeverFact** dataset – and compile the criteria previously studied for extracting factual claims [Wright et al., 2022; Kuhn et al., 2013], proposing methods of their automatic evaluation at scale.

This chapter explores claim extraction using generative methods in breadth, accumulating a set of data, automated reference-based and reference-free metrics, and proposing

[1] An interesting close parallel for this task are the single-sentence summarization methods (also known as *extreme summarization*) explored in Narayan et al. 2018 and Mao et al. 2022, fine-tuning of which we examine as one of the approaches in Section 5.3.

solvers for the task, yielding a complete scheme of automated generation of check-worthy claims and measuring their qualitative properties.

5.1.1 Contributions

1. We publish the FEVERFACT dataset (Figure 5.1) with 4.4K contextualised Wikipedia sentences and a total of 17K check-worthy claims extracted from them by annotators.
2. We propose an automated evaluation framework for claim extraction reusable for other generative fact-critical tasks, consisting of 6 metrics: Atomicity, Fluency, Decontextualization, Faithfulness, Focus and Coverage. We compile current research in the field to name the relevant metrics, as well as to find their scoring methods, combined with novel ideas where appropriate (see Section 5.4).
3. We explore generative methods for extracting FEVERFACT claims – QACG from Pan et al. 2021, LLM prompting, and LM transfer learning, publishing our tuned models.
4. We annotate a blinded sample of generated claims to validate our evaluation framework and challenge the benchmarks it produces.

5.2 FEVERFACT Dataset

In order to obtain a dedicated dataset, we utilise the **FEVER Error Correction** data published by Thorne and Vlachos 2021. Despite the dataset being designed for a different task, it releases a set of 17.5K truthful claims *directly extracted* from a sample of sentences of the 2017 Wikipedia corpus. The annotation of these claims dates back to the WF1a task of the original **FEVER** [Thorne et al., 2018a], in which the annotators were instructed to extract 2–5 claims from a source sentence, being provided with its neighbouring sentences and page title for context, as shown in Figure 5.1 – hereinafter, we refer to this whole body of text as the *FEVERFACT document*. The original source sentence from which each claim was extracted cannot be reproduced from the data, as each sentence is only represented by an integer ID pointing to a mapping file within a deleted Amazon bucket.[2]

With a simple idea, we manage to reverse-engineer this mapping. We first group the claims by the ID of their (unknown) source sentence, taking the most common Wikipedia article among each group's FEVER gold evidence. This should be the article the group was extracted from, since all the WF1a claims were annotated to be true and only contain the information from their source sentence, without any additional knowledge. We then scan this article sentence-by-sentence using a NLI model, looking for a sentence which entails the most of the claims in the group, tossing the groups where no claim is entailed by any sentence due to noise in our method. Doing so using a `nli-deberta-v3-small` [He et al., 2021, 2023; Reimers and Gurevych, 2019] pre-trained CrossEncoder yields near-ideal results – our annotation on a sample of 2% (89 sentences, 350 claims) of datapoints shows that in 94.4% of cases the *entire* group of claims could be directly extracted from the reverse-engineered source sentence, only using its neighbouring sentences and page title to resolve coreferences.

Thus we release the dataset we call **FeverFact**: 4.4K three-sentence documents (plus a page title), each annotated with a set of claims that may be extracted from its middle sentence, for a total of 17K atomic factual claims. We split it into train, dev and test sets in an 80:10:10 ratio, preventing the same page title from appearing in two different splits.

[2]As per direct communication with FEVER authors.

5.2.1 FeverFact Recall

In the wild, an ideal dataset for claim extraction would feature *all* check-worthy claims that can be extracted from a given source sentence or even from the whole FeverFact document. Such data is extremely hard to annotate due to the requirement for claim atomicity[3] – even relatively simple grammatical parallelisms such as zeugmata or compound sentences can explode the number of relations that can be extracted from the sentence.

Even so, having a large enough sample of claims matched to a source text yields a good approximation of which information within the text is check-worthy enough to be extracted.

The original annotators were tasked to extract 2–5 check-worthy claims from the source sentence, producing a median of 4 claims per source. To probe whether this number is enough, we perform Named Entity Recognition (NER) on both FeverFact claims and their source sentences using NameTag 3 [Strakova et al., 2019], finding that the recall of named entity words in FEVER claims is about 67% (taking source entities as a reference). While such a metric is on the rough side, we conclude that the FeverFact claims cover most of the information and are suitable enough to be used as a reference for reference-based claim extraction metrics (Section 5.4.2).

5.3 Claim Extraction Models

Using the FeverFact dataset to train and evaluate claim generators, we model automated claim extraction as a "one-to-many" sequence-to-sequence task [Sutskever et al., 2014] – given a single FeverFact document, predict a set of facts it claims. We experiment with prompting, transfer learning, and a NER-based baseline used on FEVER data previously in the literature.

For all models, we do not provide any distinction (such as separators) between the FeverFact source sentence and its context, feeding the whole FeverFact document (Figure 5.1) to each model's input. This is done experimentally, to test whether different models are able to learn to focus only on the information relevant to the user (in our case, arbitrarily, only the information claimed by the middle sentence) just from training or few-shot examples. If so, this primitive example of "positional" relevance of input information could motivate the collection of claim extraction datasets capturing more challenging notions of relevant information to extract, such as check-worthiness of claims scattered through a political debate or Twitter discourse [Barrón-Cedeño et al., 2023].

We experiment with the following models:

1. **Question Answering for Claim Generation (QACG)** [Pan et al., 2021] is an off-the-shelf baseline. It has been built dedicatedly to generate claims: given an input text, its named entities are extracted, fixing each entity e as an answer for a question Q to be generated. The (Q, e) pair is then converted into a declarative sentence using a QA-to-claim model, arriving at one factual claim for each of the source-text entities.

   While being a valuable baseline, the entity-centric approach has its caveats: the QACG pipeline has many steps, each a language model and therefore a point of failure – a mistake in any step propagates to the result. QACG also lacks trainable parameters for *choosing the appropriate* claims. If, for example, one wants to generate claims from a debate, it does not have mechanisms to learn to omit recurring irrelevant, entity-dense guest introductions.

[3] Each claim describing a single entity or relation; see Section 5.4.1.

2. **LLMs and few-shot learning** [Brown et al., 2020] has been recently a popular, universally well-performing solution. Models such as GPT-4 [OpenAI, 2023a] or Mistral-instruct [Jiang et al., 2023] have been shown to adapt well for similar tasks.

   We examine `gpt-4-turbo` in a 3-shot setting and fine-tune `Mistral-7b-instruct-v0.2` on our FeverFact train data using the quantised QLoRA [Dettmers et al., 2023] approach with $r = 64$, $\alpha = 32$ on 4 bits.

   The GPT-4, however, is a black box, and the open-source LLMs are computationally expensive. This motivates an examination of other methods that are reproducible on much lower resources using easy-to-obtain data.

3. **T5 transfer learning** [Raffel et al., 2020]: our small model of choice (based on preliminary experiments) is `t5-small-finetuned-xsum`.

   T5 was pre-trained on various text-to-text tasks using a *span-corruption* objective, making it adapt well to our task despite a relatively small (3.5K FeverFact documents, 13.5K claims) training size. We found that fine-tuning on a single-sentence summarisation dataset such as XSum [Narayan et al., 2018] *and then* on FeverFact yields even better results, possibly due to similar task definition.

   Two approaches were examined: first, where T5 was trained to output a concatenation of all claims in a single prediction,[4] second, where T5 was tuned to output a *single* claim per sentence (using our data as 13.5K document-claim pairs). In the prediction stage, the latter approach was coupled with *diverse beam search* [Vijayakumar et al., 2018] decoding to generate an arbitrary number (k) of single-sentence claims per text, using k beam groups and a diversity penalty of 1. Its competitive results are particularly encouraging, as single-sentence summarisation data is easily available in other settings and languages – XLSum [Hasan et al., 2021] alone features data in 45 languages.

5.4 Evaluation Framework: Claim Metrics

To evaluate our models, we compile claim quality criteria introduced in other research. Each predicted claim is checked to be *faithful*, *fluent*, *atomic* and *decontextualised*. A set of claims extracted from the same document is measured to *focus* solely on its relevant information and *cover* all of it. While the criteria above can be checked by human grading as in Wright et al. 2022, research such as Wang et al. 2020 and Koto et al. 2020 already proposes ways of how to evaluate some of the qualities automatically, alleviating the need for human grading of model outputs. In this section, we attempt to compile known evaluators for our task, reducing each metric to an already well-explored NLP challenge, and then validate our metrics against real human annotations on generated claims.

5.4.1 Reference-free Evaluation Metrics

For the reference-free metrics which score each claim on its own, we use the criteria named by Wright et al. 2022: Atomicity, Fluency, Decontextualization, and Faithfulness, adapted from Kuhn et al. 2013's AIDA (Atomic, Declarative, Independent, Absolute). To allow automated evaluation at scale, we find the following reductions to known NLP tasks to replace human grading:

[4] The claims were then separated using PySBD [Sadvilkar and Neumann, 2020].

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

   While real-world factual claims are not often atomic in the strict sense (i.e., claims are typically more convoluted than "A is B" or "C does D"), breaking a more complex factual statement into a set of atomic claims trivialises inference on top of such claims and allows certain explainability [Dammu et al., 2024], such as which parts of a complex statement contradict which facts.

   We propose the following scheme of atomicity classification of claim c, reducing it to a Relationship Extraction task:

   $$A(c) := \begin{cases} 1 & \text{if } |RE(c)| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

   where the relation-extraction result is interpreted as a set of undirected relations $RE(c) = \{\{s_1, t_1\}, \ldots\}$, in order to avoid counting symmetrical relationships like $(Trump, president\_of, USA)$ and $(USA, governed\_by, Trump)$ twice.

   For the RE solver, we recommend using REBEL [Huguet Cabot and Navigli, 2021] due to its manageable size and end-to-end approach – other models may rely on entity-pair extraction and classification [Yamada et al., 2020].

   We also experimented with a non-binary metric $A'(c) = \frac{1}{\max\{1, |RE(c)|\}}$, but during our experiments, non-atomic model outputs were quite rare, so zeroing out the A score worked best for averaging across a large number of claims.

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

   The task of fluency, also referred to as *grammaticality*, is well studied in the literature, with recent research modelling it as grammatical error detection (GED) and correction (GEC) tasks.

   The available techniques range from simpler ridge regression models based on linguistic features [Heilman et al., 2014] through using syntactic log-odds ratio (SLOR) [Kann et al., 2018] or perturbing claim words and characters to find local optima in the output probability using a language model such as GPT-2 [Yasunaga et al., 2021] to prompting an LLM to obtain a model-inferred score using few- or zero-shot learning [Fu et al., 2024].

   The best-performing approach we studied revolves around CoEdIT [Raheja et al., 2023] GEC model, coupled with Scribendi score [Islam and Magnani, 2021] to rate the improvement between each claim and its CoEdIT correction. Scribendi score combines perplexity scores and Levenshtein distance, yielding -1 for bad correction, 0 for no improvement, and $+1$ for correction improving the claim. While this metric is originally designed for the evaluation of GEC systems, it performs well on rough fluency rating, looking for zero or negative improvement in CoEdIT corrections:

   $$G(c) := \begin{cases} 1 & \text{if Scribendi}(c, \text{CoEdIT}(c)) \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

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

   Choi et al. 2021 proposes decontextualisation as a text-to-text task, training a T5 to receive context and a sentence on its input, outputting the decontextualised sentence by resolving pronouns into proper nouns and relative terms into absolute according to the given context.

As with atomicity, a strict binary classification worked best, as non-decontextualised model outputs were rare. For context, we use the full FEVERFACT document d:

$$D(d,c) := \begin{cases} 1 & \text{if } \mathrm{T5_d}(d,c) = c \\ 0 & \text{otherwise} \end{cases}$$

where $\mathrm{T5}_d(d,c)$ denotes the output of a $\mathrm{T5_{large}}$ model trained by Choi et al. 2021.

4. **Faithfulness** – *does the claim contain only information consistent with its source?*

   Faithfulness has been extensively studied to detect hallucination in text-to-text tasks. ALIGNSCORE claimed state of the art in Zha et al. 2023, looking for optimum alignment of output and input parts in terms of a ROBERTA [Liu et al., 2019] classifier with a *unified alignment function*. While compact (125M parameters in the *base* version), it outperforms metrics based on GPT-4 that are orders of magnitude larger [Zha et al., 2023].

$$Faith(d,c) := \mathrm{ALIGNSCORE}_{base}(d,c)$$

5.4.2 Reference-based Evaluation Metrics

The evaluation metrics introduced above do not rely on any gold data for their reference. But what if we need to evaluate the whole group of claims, to see if it captured all the information it should and nothing more?

The notion of what a good set of claims is varies task-to-task, so engineering rules may do more harm than good. Let us therefore use examples, as Koto et al. 2020 suggests in their FFCI framework for interpretable summarisation metrics, measuring *Focus* and *Coverage* to see whether the model extracts what a human would. Besides the predicted set of claims C, which serves as the input for multi-claim metrics, assume a set of gold claims G extracted from the same document – in FEVERFACT, those would be the full sets of claims (green in Figure 5.1) obtained from annotation.

5. **Focus** – *what is the proportion of gold (relevant) information among all the information listed in the generated claims?*

   The metric is analogous to the concept of *precision*: $\frac{|true\ positives|}{|positives|}$, but rather than exact matching, we seek a measure of semantic overlap. FFCI uses QAGS [Wang et al., 2020], which uses a Question Generation model (QG) to formulate questions in natural language based on all $|C|$ predicted claims. The questions are then answered twice using a Question Answering (QA) model, with knowledge from (i) the predicted claims and (ii) the gold claims written by a human. 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. In our experiments, this metric was too noisy, with QG, QA and aggregation faults all propagating into the final result.

   To propose a simpler method, we exploit the fact that gold claims in G can be concatenated into a mock-document. Since gold claims are decontextualised by annotation, this preserves their meaning, reducing Focus into $|C|$ independent tasks of deciding Faithfulness:

$$Foc(G,C) := \frac{1}{|C|}\sum_{c\in C} Faith(\mathrm{concat}(G), c)$$

We also encourage experiments with *claimwise focus*, computing the single probability $Foc(G, \{c\})$ for each predicted claim c separately, to see which exact claim in C should be extracted according to G and which should not.

6. **Coverage** – *what proportion of gold (relevant) information from the source is featured in C?*

   Can be simply adapted from Focus:

$$Cov(G, C) := Foc(C, G)$$

   Koto et al. 2020 proposes this trick, and while our underlying Foc method differs, the argument swap still yields the proportion of gold information extracted into C. Whether the claims in C are non-decontextualised by a faulty prediction and could influence each other's meanings upon concatenation can be checked for using the metric from Section 5.4.1, marking the claims to toss, or leaving them in if the level of noise is tolerable. We went with the latter as $> 94\%$ of claims were decontextualised upon human grading (Table 5.2).

7. **Redundancy**[5] – *What is the proportion of duplicate information among generated claims?*

   The definitions of Foc and Cov inspire an elegant estimate of *model redundancy* – if $|C|$ claims are generated by a one-to-many seq2seq model, what proportion of them can be expressed from the *other* model outputs:

$$Red(C) := \frac{1}{|C|} \sum_{c \in C} Faith(\text{concat}(C \setminus c), c)$$

5.4.3 F_{fact}-value

As our experiments in Section 5.5 suggest, Foc and Cov indeed behave like Precision and Recall metrics in the sense of their mutual tradeoff – if a naïve model scores very high in one (such as QACG covering *every part* of the FEVERFACT document), a significant decrease can be observed in the other.

In our experience, the reference-free single-claim metrics (Section 5.4.1) did not pose a tough challenge to modern NLP methods, most of their corresponding mean scores in both Table 5.1 and Table 5.2 being very high, raising a concern they may be too-easy benchmarks nowadays. Provided with the gold annotations for each source document, we therefore suggest the F_{fact} **score** – the harmonic mean of Foc and Cov – as the primary aggregate metric for comparing claim extraction systems:

$$F_{fact}(G, C) := \frac{2 \cdot Foc(G, C) \cdot Cov(G, C)}{Foc(G, C) + Cov(G, C)}$$

5.5 Results

After training the models described in Section 5.3 on FEVERFACT, we had each model extract a set of claims from each of 444 FEVERFACT test documents. We evaluated the claims using the automated metrics described in Section 5.4. To validate the metrics, we also annotated a subset of 9% (40) FEVERFACT test documents (and 1110 claims extracted from them by the models in total), using a custom annotation platform.

[5] Unlike Focus and Coverage, we did not validate this Redundancy metric with annotations, but as its validity stems from the same principles as that of Foc and Cov, we include it for added insights.

Model	Atomicity	Fluency	Decontext.	Faith.	Focus	Coverage	F_{fact}	Redundancy
qacg	0.89	0.69	0.70	0.88	0.20	0.67	0.30	0.44
gpt-4-turbo-3-shot	0.92	0.70	0.77	**0.99**	0.21	**0.81**	0.34	**0.14**
qlora-mistral-instruct	0.95	0.75	0.80	0.95	**0.58**	0.63	**0.61**	0.19
t5_sm_diverse_7_beams	0.95	0.74	0.80	0.91	0.55	0.58	0.56	0.59
t5_sm_multi-claim	**0.96**	**0.76**	**0.82**	0.95	**0.58**	0.51	0.55	0.54
Evaluation method	*REBEL* *$\lvert rel \rvert \leq 1$*	*CoEdIT* *+Scribendi*	*T5* *Decontext*	*Align-Score*	*Concat AlignS.*	*Concat AlignS.*		
Validation against human	*0.96*	*0.80*	*0.86*	*0.92*	*0.23*	*0.22*		
Validation method	*F_1 – higher is better*				*Root mean squared error – lower is better*			

Table 5.1: Automated claim metric averages across model-generated claims on the FEVERFACT test set. The best value for each metric is in bold. Model choices and training procedures are described in Section 5.3, claim quality metrics in Section 5.4. Automated scoring methods and their validation against human grading (Table 5.2) are in italic.

Model	Atomicity	Fluency	Decontext.	Faith.	Focus	Coverage	F_{fact}
qacg	0.99	0.85	0.91	0.76	0.19	0.60	0.28
gpt-4-turbo-3-shot	0.98	**0.97**	**0.96**	**0.93**	0.24	**0.79**	0.37
qlora-mistral-instruct	**1.00**	0.96	**0.96**	0.90	**0.60**	0.69	**0.64**
t5_sm_diverse_7_beams	0.99	0.89	0.95	0.79	0.51	0.62	0.56
t5_sm_multi-claim	**1.00**	0.91	0.94	0.88	0.47	0.50	0.49
Inter-annotator agreement (sample of $\sim$ 23% annotated claims, $\leq$ 5 annotators per claim, at least 2)							
Krippendorff's α	*0.27*	*0.41*	*−0.01*	*0.53*	*0.75*	*0.64*	
GWET's AC1	*0.95*	*0.86*	*0.85*	*0.83*	*0.80*	*0.66*	
%-agreement	*0.87*	*0.86*	*0.97*	*0.84*	*0.88*	*0.82*	

Table 5.2: Blinded human annotation averages across a sample of 1110 claims generated from 40 (9%) FEVERFACT test documents. Annotation of reference-free metrics (left) was done using grading scales adapted from Wright et al. 2022, binarising the best grade to 1, others to 0. Reference-based metrics (right) were annotated using a checkbox interface over gold and predicted claims. The best value for each metric is in bold. Inter-annotator agreement results are in italic. Krippendorff's α is left for completeness, albeit very inappropriate for tasks with such class imbalance [Wright et al., 2022]; Gwet's AC1 [Gwet, 2010] is the more appropriate agreement metric for our task.

5.5.1 Model Comparison

In Tables 5.1 and 5.2 we use the automated and human-annotated metrics to benchmark our models, on the full FEVERFACT test set and a sample of 40 documents, respectively.

The benchmarks reveal that all models score high in Atomicity, Fluency, Decontextualization, and Faithfulness reference-free metrics – confirmed by the human annotations in Table 5.2, these do not appear to be the challenges modern NLP claim-extractors struggle with.

The reference-based metrics of Focus and Coverage are where significant differences can be found and tradeoffs can be seen – models with highest Coverage focus too little and vice versa. Taking their harmonic mean, F_{fact}, the fine-tuned models `qlora-mistral-instruct` and `t5_sm` take a significant lead, showing that the models were able to learn that the FEVERFACT claims come from the middle sentence on input – the information annotators were tasked to focus on in the original FEVER annotation experiment [Thorne et al., 2018a]. This suggests these models could be tuned for trickier notions of check-worthiness, such as those studied by Barrón-Cedeño et al. 2023.

The T5 coupled with *diverse beam search* decoding shows promising results, despite being essentially an abstractive summarisation model. Its redundancy is the highest (diverse outputs only assured through a decoding strategy), but not a clear outlier. We find this very encouraging for further use of diverse beam search in settings where only one claim per source document is annotated, such as in AVeriTeC-DCE [Deng et al., 2024], to produce multiple interesting results. Datasets like XLSum [Hasan et al., 2021] with an objective highly similar to claim extraction and good availability of summarisation models across languages make it an accessible choice with competitive performance.

5.5.2 Metric Validation

To validate our evaluation framework, we annotated the claim properties on grading scales with instructions adapted from Wright et al. 2022 and Focus and Coverage using a simple checkbox interface.

The human annotations confirm that the reference-free methods (Section 5.4.1) are easy for today's models. They also preserve the leaderboard of models based on F_{fact}, the hardest metric, requiring an understanding of which claims are relevant in FeverFact data and punishing greedy approaches. We evaluate the reference-free metric estimation methods using the F_1 score, chosen due to the imbalance of classes on real generated claims. For faithfulness and fluency, we binarise the human-annotated grades (the highest grade to 1, others to 0), to be directly comparable with automated metrics – AlignScore used for Faithfulness was simply rounded to 0 or 1 (threshold 0.5) for this experiment. All metrics score above 80% F_1, testifying to their soundness and usability at scale in scenarios where a good approximation is sufficient.

We proceed to validate our Focus and Coverage metrics against human annotation, using the root mean squared error to obtain values around 0.22, which is also encouraging for further use. To see whether this value is not accidental (correct proportion, but erroneously distinguished gold claims), we also measure the *claimwise* Foc and Cov to see the per-claim contribution to metrics as probabilities $Foc(G, \{c\})$ and $Cov(\{g\}, C)$, as shown in Section 5.4.2. We compare them to claimwise annotations and assess their quality using a Brier score as suggested in Rufibach 2010. We measure 0.15 for Foc and 0.12 for Cov (lower Brier score is better), in other research sometimes interpreted [Fernandes Salvador, 2011] as a "superior" agreement.

5.5.3 Inter-Annotator Agreement

The inter-annotator agreement study is presented in Table 5.2. Due to missing annotations, our first choice was the Krippendorff's α coefficient [Krippendorff, 2011]. Its results are, however, very low on our annotations, even reaching negative values – the cause of this is the very high level of class prevalence in our annotations,[6] overly increasing the chance of accidental agreement that Krippendorff's alpha targets to punish. This is a known pitfall of Krippendorff's alpha, also encountered in the Wright et al. 2022 paper which proposed the reference-free metrics. To overcome it, we also report the Gwet's Agreement Coefficient 1 (AC1) [Gwet, 2010] designed to be less sensitive to trait prevalence. The AC1 coefficient is high in all cases, supporting the conclusion that our annotation process was reliable. For completeness, we also report the %-agreement – the proportion of labels on which all annotators agree [Artstein, 2017].

[6] Prevalence of the most common class in faithfulness, fluency, atomicity and decontextualisation grading is 0.84, 0.9, 0.98, and 0.91, respectively.

5.6 Conclusion

This chapter examined the problem of automated claim extraction in its full scope: from data gathering (publishing FeverFact) through model training (suggesting low- and high-resource options), benchmarking (publishing a Python framework for automated evaluation) to its validation with annotators. The hardest and most appropriate score for claim extraction is the F_{fact} score, requiring gold claims for reference and measuring the balance between the precision-like Focus and recall-like Coverage.

Interestingly, we have shown that the task of claim extraction can be substantially addressed using single-sentence abstractive summarisation methods with a diverse beam search decoding strategy, which is particularly useful for low-resource languages and environments.

A central challenge to claim extraction is the notion of claim *check-worthiness*, now explored as a classification task [Barrón-Cedeño et al., 2023]. Our work models a simplified version of it, training the models to focus only on the middle sentence claims in the manner that the FEVER annotators did. Adapting existing data such as CLEF-CheckThat! datasets for text generation, rather than classification, could capitalise on our findings to train generative models which focus only on check-worthy claims. The evaluation metric framework can also yield additional insights into any system for knowledge-intensive tasks, especially where multiple gold answers can be found or synthesised, such as Summarization or Retrieval-Augmented Generation.

Chapter 6

RAG-Based Fact-Checking Pipeline

Research Questions

RQ7 Does a single LLM request per claim, with a prompt augmented by retrieval, yield competitive performance in automated fact-checking?

RQ8 Does the RAG fact-checker still perform well when built on top of open-weights LLMs?

RQ9 Can a text-based RAG fact-checking pipeline be extended to image-text claims while remaining competitive?

6.1 Introduction

We present 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 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 chapter covers three submissions of this pipeline to consecutive shared tasks. In AVeriTeC [Schlichtkrull et al., 2024b], we scored third on the leaderboard with an AVeriTeC test set score of 50.4%, using GPT-4o as the generative component. In AVeriTeC 2 [Schlichtkrull et al., 2024a], the challenge was reformulated: systems were required to use exclusively open-weights models under a fixed compute budget of a single 23 GB A10 GPU and 60 seconds per claim on average. We adapted our pipeline by replacing GPT-4o with a locally hosted Qwen3-14B instance – and topped the AVeriTeC 2 test leaderboard with a significant margin. Finally, in AVerImaTeC [Cao et al., 2025], the task was extended to image-text claims whose veracity depends not only on the textual statement but also on the images attached to it. We extended our pipeline with a Reverse Image Search (RIS) retrieval module and a multimodal LLM, placing 3rd in the AVerImaTeC shared task.

Section 6.3 describes the core text-based pipeline design. Sections 6.4 and 6.5 cover the AVeriTeC 2024 experiments. Section 6.6 presents the AVeriTeC 2 open-source adaptation. Section 6.7 describes the AVerImaTeC extension to image-text fact-checking.

[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.

6.2 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, Application Programming Interface (API) search (producing a knowledge store), sentence retrieval, Question-and-answer (QA) generation, QA reranking, per-QA claim classification and label aggregation, achieving an overall AVeriTeC test set score of 11%.

2. **FEVER Shared Task** [Thorne et al., 2018b], 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 NLI, improving its modules in a decoupled manner and scoring well above 60% in a similarly computed FEVER score [Thorne et al., 2018a] on this data.

3. **Our previous research** on fact-checking pipelines [Ullrich et al., 2023; Drchal et al., 2024] 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. **RAG for Knowledge-Intensive Tasks** [Lewis et al., 2020] takes this a step further, leveraging 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.

6.3 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.

6.3.1 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., 2024], 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 ω chunks, setting the ω 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 6.3.1 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: Maximal Marginal Relevance (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' 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].

6.3.2 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 an LLM, instructing it to output both evidence and the label in a single step, as a chain of thought. We rely on structured JavaScript Object Notation (JSON) 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 6.3.2) 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 a 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 6.3.2) 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 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

[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.

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., 2020] 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 NLI (see section 6.4.3).

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 6.1 and 6.2; for our submission in the AVeriTeC shared task, GPT-4o was used.

6.4 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 6.3.2) 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 6.3.2).

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 6.2 and 6.1) 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 6.1) for the fact-checking task.

6.4.1 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 BERT [Devlin et al., 2019] 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.

6.4.2 Multi-evidence classification

The multi-evidence approach is to fine-tune a 4-way 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 prefinetuned 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.

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

[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`

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].

6.4.3 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 6.3.2, 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 6.1.

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 6.1 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 6.2).

6.4.4 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 6.4.2), 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.

6.5 AVeriTeC 2024: 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 the retrieval module is fixed throughout the experiment (a full scheme described in section 6.3.1), various Evidence & Label generators and classifiers are compared in Table 6.1, 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

[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 6.1: 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 6.2: 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 6.2 thresholded at 0.25. Full-featured pipelines use the all the improvement techniques introduced in section 6.3, while the simplified pipeline omits the dynamic few-shot learning, answer-type-tuning and Likert-scale confidence emulation described in section 6.3.2

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 6.2 and in greater detail in [Schlichtkrull et al., 2024b]. We use the provided AVeriTeC scoring script to calculate the values for Table 6.2, using its EvalAI blackbox to obtain the test scores without seeing the gold test data.

The latter experiments shown in Table 6.2 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 6.3.2), 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.

6.5.1 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].

6.5.2 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 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

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

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.

6.6 AVeriTeC 2025: Open-Source Adaptation

In AVeriTeC 2 [Schlichtkrull et al., 2024a], the challenge was reformulated: all systems were required to use exclusively open-weights models, constrained to a fixed compute budget of a single 23 GB A10 GPU and an average of 60 seconds per claim. This raised the central question of the task – can the fact-checking process be automated in a way accessible to the masses, or is its quality conditioned on business-owned blackbox models or prohibitive computational resources?

We adapted the AIC CTU system from Section 6.3 directly for this setting. The two main differences from the 2024 submission are the omission of the BM25 knowledge store pruning step – not allowed in AVeriTeC 2, since vector stores were required to be precomputed independently of the claim text – and the replacement of GPT-4o with a locally hosted open-weights LLM. All other design choices (embedding model, chunk size, FAISS index, MMR parameters) remain identical to Section 6.3.

6.6.1 Model and parameter choices

`Ollama` wrapper around `llama.cpp` is used as the LLM engine within the AVeriTeC 2 test environment due to its robustness and ease of deployment.

`Qwen3-14b` [Yang et al., 2025] is the LLM used to produce evidence and labels. We also let it generate its own `<think>` sequences, although further experimentation (Table 6.4) suggests that the thinking tokens do not justify the costs of their prediction, performing on par with using only the evidence & label outputs for the chain of thought.

Our pipeline design uses a single LLM call per claim, which allowed us to fit the generously-sized 14B variant of Qwen3 within the time limit on the Nvidia A10 GPU.

6.6.2 Results

Our system tops the AVeriTeC 2 test leaderboard (Table 6.3) with a significant margin in the decisive new AVeriTeC score. This came as a surprise to its authors: neither the old hu-METEOR-based AVeriTeC score nor the dev-leaderboard available during the development phase (where our system scored 4th) suggested its supremacy.

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

Table 6.3: AVeriTeC 2 shared task system leaderboard as shared by organizers, listing new Ev^{2}R-recall-based [Akhtar et al., 2024] and legacy hu-METEOR AVeriTeC scores. Evaluated using AVeriTeC 2 test set. Best scores are bold.

6.6.3 Why does the system perform well?

The main reason, in our experience, is the large **context size** we opt for. Even the AVeriTeC 2 baseline 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. Combining sources of total length of as much as 60K characters[14] on the model input yields highly competitive results, leveraging the LLM's 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 previous LLM generations at long sequence processing.

6.6.4 Scoring change impact

The new AVeriTeC score based on Ev^{2}R-recall [Akhtar et al., 2024] estimates the proportion of correctly fact-checked claims in all claims – just like the old hu-METEOR-based AVeriTeC score – but the underlying methods differ. Most importantly, an LLM-as-a-judge approach is now used instead of symbolic evidence comparison. The rise of our system from 3rd place in the AVeriTeC shared task to 1st place in AVeriTeC 2 without any major system change[15] can therefore also be partly attributed to the change in scoring method. The old scoring method was, for example, found to be prone to noise and not

[14] Around 33 standard pages. This follows from our parameter choices in Section 6.6.1: 10 sources per claim, each with ∼2048 characters of embedded text and additional ∼4096 characters of context.

[15] Despite scaling down.

robust against evidence duplication [Malon, 2024], which was a known exploit to boost evidence recall.

The discrepancy between old and new AVeriTeC score in Table 6.3 motivates a further study of how the new score behaves, for example using the prediction files from the 2024 systems, where both scores are available.

6.6.5 LLM impact

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

Table 6.4: Ablation study on LLM choice and `<think>`-token impact on AVeriTeC 2 dev-score. Pipeline design (Figure 6.1), retrieval results, system and user prompts are fixed. Evaluated using an on-premise Ev^2R scorer with Ollama-hosted Llama3.3-70B as a judge.

In 2024, we experimented with then-available versions of GPT-4o and Llama3.1-70B and found the open-source Llama to perform encouragingly well, despite the cumbersome model size and the need for its quantization [Ullrich et al., 2024]. For AVeriTeC 2, we chose the most recent open-weights LLM at the largest parameter count fitting within the compute budget: Qwen3 at 14B parameters [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. Table 6.4 shows that both Qwen3 settings perform on par with GPT-4o generation from 2024, validating the model choice. The thinking tokens, while producing legitimate-looking writeups of the fact-checking reasoning process (see Appendix B.4), do not improve the AVeriTeC score in the ablation study. We therefore suggest disabling this feature in future reproductions in favour of faster prediction time – the 54 s runtime in Table 6.3 was produced with thinking *enabled*, so disabling it should provide more headroom before the time limit.

6.7 AVerImaTeC: Extending to Image-Text Fact-Checking

With public discourse moving increasingly to social media, the task fact-checkers face often goes beyond just text and language. An important example are image-text claims, whose veracity depends not only on the textual statement itself but also on the images accompanying it – whether they are authentic or edited, and whether they are presented in the right context.

The AVerImaTeC shared task [Cao et al., 2025] targets this setting, collecting hundreds of reference image-text fact-checks from human annotators. We extend the pipeline from Sections 6.3 and 6.6 with an image-based retrieval module, using a single query to a multimodal LLM per claim and a single reverse image search (RIS) request per attached image. Our system places 3rd in the AVerImaTeC shared task.

6.7.1 Image-text system design

Our pipeline, depicted in Figure 6.2, extends the two-module RAG scheme from Section 6.3 into a dual-retrieval architecture with three modules:

i. **Text-based retrieval module** – identical to Section 6.3, with parameters $k = 20$, $l = 7$, $\lambda = 0.8$:

   1. **Vector store** is produced for each of the AVerImaTeC datapoints in advance using the scheme from Section 6.3: the provided text-only[16] AVerImaTeC knowledge store is chunked into 2048-character segments[17], and each is embedded using `mxbai-embed-large-v1` [Li and Li, 2024; Lee et al., 2024].
   2. **Similarity search** is performed using exact k-NN search via `FAISS` [Douze et al., 2024; Johnson et al., 2019], with $k = 20$ nearest neighbours.
   3. **Maximal marginal relevance** [Carbonell and Goldstein, 1998] reranking down to $l = 7$ results is applied to diversify the search results ($\lambda = 0.8$ in favour of similarity to the claim).

ii. **Image-based retrieval module** is invoked separately for each image attached to the AVerImaTeC claim – if a claim contains n images, n separate sets of results are produced:

   1. **Reverse image search (RIS)** is performed using `Google Lens`[18] via Serper API[19] to produce a set of (up to 30) results – each assigned a webpage URL and a *thumbnail* of an image within that webpage similar to the given claim image.
   2. **Scraping**: each RIS result is scraped using Firecrawl API[20], producing LLM-friendly markdown. We retain only the thumbnail that triggered the RIS result, as it has stronger guarantees of being similar to the claim image.
   3. **Result filtering** – to maintain the evidence principle [Glockner et al., 2022], we filter out evidence published after the claim was originally stated, using the Htmldate [Barbaresi, 2020] library to estimate publication dates.

      Many RIS results were scraping-protected (notably Facebook and Instagram posts), yielding empty results; these are discarded as well. Finally, to fit a single-digit source identifier (used in module iii) and keep the generation prompt uncluttered, we retain at most 9 of the remaining results.

iii. **Evidence, label, and justification generation module**:

   1. **System prompt** is composed of results from both the text- (i.) and image-based (ii.) retrievers, referred to as **text-related sources** and **image-related sources** respectively. Sources are assigned numerical IDs: 1–9 for text sources, 11–19 for sources related to the 1st claim image, 21–29 for the 2nd, and so on. For image sources, we include only their scraped text and a note that it was

[16] AVerImaTeC also includes image-text and image-only knowledge stores, but since these were (as of Feb 2026) not marked with a source URL or other real-world identifier, we dropped these as inappropriate to be referred to as sources.

[17] The chunks do not overlap, and are annotated with context before and after in their metadata, as described in more detail in Section 6.3.

[18] `https://lens.google.com/`

[19] `https://serper.dev/`

[20] `https://firecrawl.dev/`

published alongside an image similar to the i-th claim image – we intentionally omit the thumbnail itself to avoid overwhelming the multimodal LLM with easy-to-confuse image inputs. These sources, the task description, formatting instructions, and few-shot examples (iii.2) are serialized into a single system prompt; its full text is in Appendix B.3.

2. **Few-shot examples** are retrieved from the AVerImaTeC train set using BM25 [Robertson et al., 1995] and appended to the system prompt to make the LLM adhere to the evidence format expected by AVerImaTeC annotators.

3. **Multimodal user message** consists of the claim text followed by base64-encoded claim images, passed to the LLM to generate evidence, label, and justification.

4. Upon **parsing** the LLM outputs, evidence items referencing an image-related source are augmented with the base64-encoded thumbnail (ii.1) of that source, to facilitate comparison with human-annotated image evidence.

5. **AVerImaTeC format matching** – as in prior years, our system outputs evidence as QA pairs. In AVerImaTeC, however, this format is phased out: the main score (Table 6.5) is now based on comparing self-contained declarative evidence texts with pointers to relevant images. To match this design without an extra LLM call, we concatenate the question and answer into a self-contained evidence string, appending `[IMG_1]` (referring to the source thumbnail) when an image-related source is cited.

6.7.2 Results and Analysis

System	Question Score	Evidence Score	Verdict Accuracy	Justification Score
HUMANE	0.89	0.54	0.55	0.56
ADA-AGGR	0.37	0.46	0.54	0.43
AIC CTU (ours)	*0.81*	*0.33*	*0.35*	*0.30*
XxP	0.39	0.27	0.26	0.20
teamName	0.66	0.23	0.26	0.22
REVEAL	0.63	0.28	0.24	0.13
fv	0.29	0.16	0.16	0.13
Baseline	0.55	0.17	0.11	0.13

Table 6.5: System leaderboard showing performance metrics on AVerImaTeC test-split. Our system is in *italics*.

The final AVerImaTeC leaderboard is shown in Table 6.5. Our system achieves a combined verdict score[21] of 0.35, a near-SOTA question score of 0.81, an evidence score of 0.33, and a justification score of 0.3. Metrics are based on Ev^{2}R [Akhtar et al., 2024] recall scores with LLM as a judge.

[21] Proportion of claims with a correct verdict *and* an evidence score of at least 0.3; see [Cao et al., 2025].

While our system does not reach the very state of the art, it significantly outperforms the iterative agentic baseline [Cao et al., 2025] and the majority of other systems across the board. To reveal directions for future improvement, we study its main pitfalls using the leaderboard metrics and our own reproductions of AVerImaTeC dev-split metrics.

Bottlenecks

Evidence format	Question Score	Evidence Score	Verdict Accuracy	Justification Score
Answer only	**0.86**	0.27	0.31	0.28
Question + Answer	*0.84*	*0.33*	***0.39***	*0.31*
Declarative evidence	0.82	**0.35**	0.38	**0.32**

Table 6.6: Ablation study on evidence format (Section 6.7, module iii). The scheme used in the final submission is in italics.

The main bottleneck appears to be our system's *evidence score*, computed using $\mathrm{Ev}^2\mathrm{R}$ recall. Our near-SOTA question score of 81% is encouraging, but the weakness in evidence score propagates to verdict and justification scores as well.

Part of this problem can be attributed to our legacy evidence format geared towards the AVeriTeC and AVeriTeC 2 shared tasks – evidence is generated as a QA pair, whereas the AVerImaTeC evaluation scheme expects a self-contained declarative sentence with pointers to relevant images.

Table 6.6 compares three approaches to this mismatch. In the first, we disregard it and list only the generated answers as AVerImaTeC evidence. In the second (our submitted system), we concatenate the question and answer and append an `[IMG_1]` tag and base64 thumbnail when an image-related source is cited. In the third, we directly prompt the LLM to generate self-contained declarative evidence with image pointers ("declarative evidence"). Although this third approach is experimental and has glitches (malformed image pointers, `[IMG_1]` appearing in justification and questions), it surpasses our submitted approach by 2% in evidence score – suggesting it is the right direction before prompt polishing.

A second bottleneck is our uncertain handling of image evidence: the "Answer only" approach, which uses *no* `[IMG_1]` tags, scores surprisingly close to the more advanced variants. This is concerning given that [Cao et al., 2025] reports 53.9% of AVerImaTeC evidence requires reverse image search annotation. Any mismatch between how our system presents image sources and what the AVerImaTeC evaluator expects could have a large impact on the final score and warrants further investigation.

Cost Analysis

The scheme from Section 6.7 uses one RIS request per claim image (the vast majority of AVerImaTeC claims feature exactly one image). Using Serper, this costs 3 credits, totalling \$0.003 at the least-discounted bulk pricing (\$50 for 50K credits).

Markdown scraping via Firecrawl costs \$0.006 per page at the hobby tier (20,000 free scrapes for education emails). In the worst case (multiple images per claim, each with 9 scrapeable RIS results and no discount), this amounts to $0.05 per image. A free alternative such as the Trafilatura library – used to produce the AVerImaTeC offline knowledge stores – can replace Firecrawl without noticeable quality loss.

Generation was computed using the OpenAI Batch API with GPT-5.1. On average, 11K input tokens and 1,150 output tokens were used per AVerImaTeC claim, at a mean cost of $0.013 per claim.

6.8 Conclusion

This chapter describes the development of a RAG pipeline for automated fact-checking and its application to three consecutive shared tasks. The pipeline's core strength is its simplicity: modular, decoupled components that leverage the capabilities of an LLM rather than complex bespoke engineering, making each year's adaptation a targeted swap rather than a redesign.

In AVeriTeC 2024, the text-based pipeline achieved a test score of 50.4%, placing third, using GPT-4o as the generative component. In AVeriTeC 2, we showed that the same design translates directly to an open-source setting: replacing GPT-4o with a locally hosted Qwen3-14B sufficed to top the leaderboard, demonstrating that competitive automated fact-checking does not require proprietary blackbox models. In AVerImaTeC, we extended the pipeline to image-text claims by adding a reverse image search retrieval module and a multimodal LLM, placing 3rd with a combined verdict score of 0.35 and a near-SOTA question score of 0.81.

We release the Python codebases for all three submissions to facilitate further research and applications, either as strong baselines or for use alongside human fact-checkers.

6.8.1 Future Works

1. Integrating a live search API [Malon, 2024] as the text retriever to achieve real-world generalization beyond offline knowledge stores
2. Re-examining the Likert-scale confidence rating (Section 6.3.2) for a more fine-grained means of expressing label uncertainty
3. Exploring evidence generation in the form of declarative sentences rather than Question-Answer pairs to see whether it improves fact-checking performance – initial AVerImaTeC results (Table 6.6) are encouraging
4. Exploring RAG-tuned LLMs [Menick et al., 2022] for more reliable source citation
5. Studying the key differences between the legacy hu-METEOR and new Ev^{2}R scoring methods using the available AVeriTeC system predictions, to reveal what system behaviours each method rewards and penalises
6. Investigating discrepancies between how our system presents image-text evidence and what the AVerImaTeC evaluator expects (Section 6.7.2), as even minor format mismatches appear to have an outsized impact on scores
7. Addressing RIS engine robustness – Google Lens often returns zero results for older claims or graphically explicit images; swapping to a more robust provider or adding an agentic fallback to text-only retrieval when RIS yields nothing would improve coverage

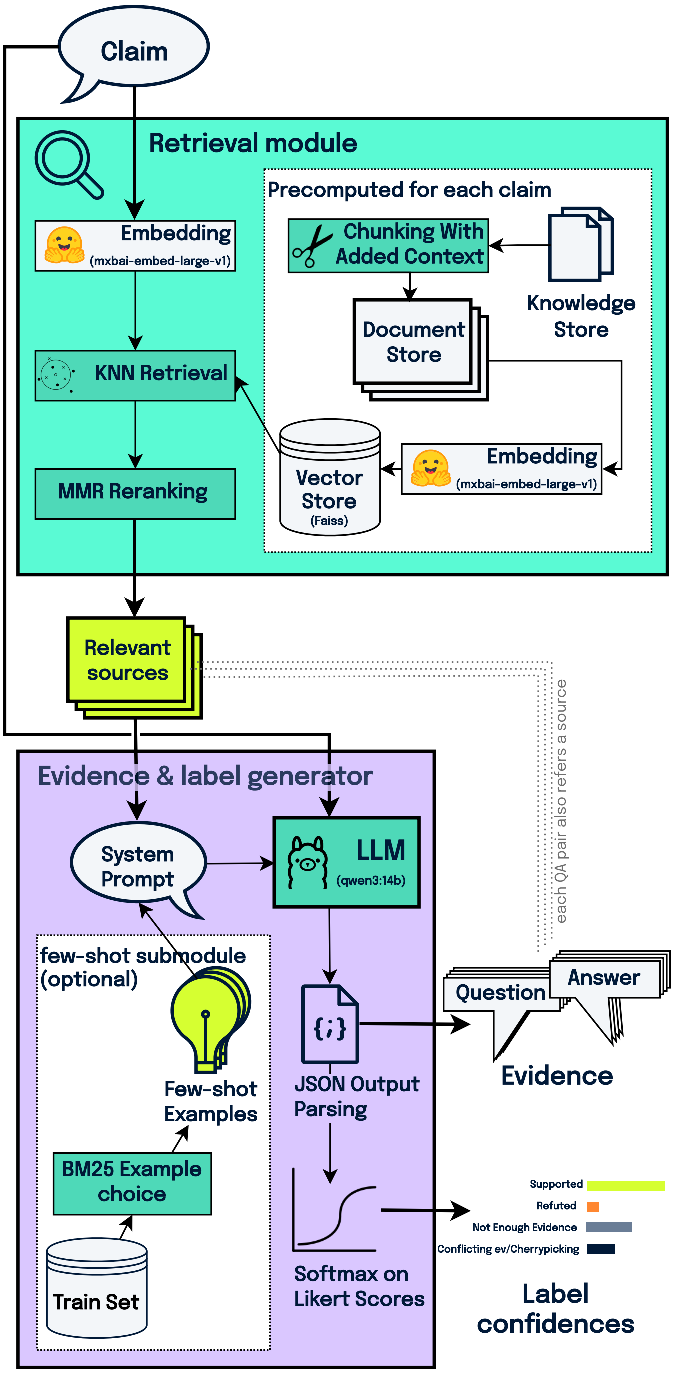

Figure 6.1: Our fact-checking pipeline, submitted to the AVeriTeC [Ullrich et al., 2024] and AVeriTeC 2 [Schlichtkrull et al., 2024a] shared tasks.

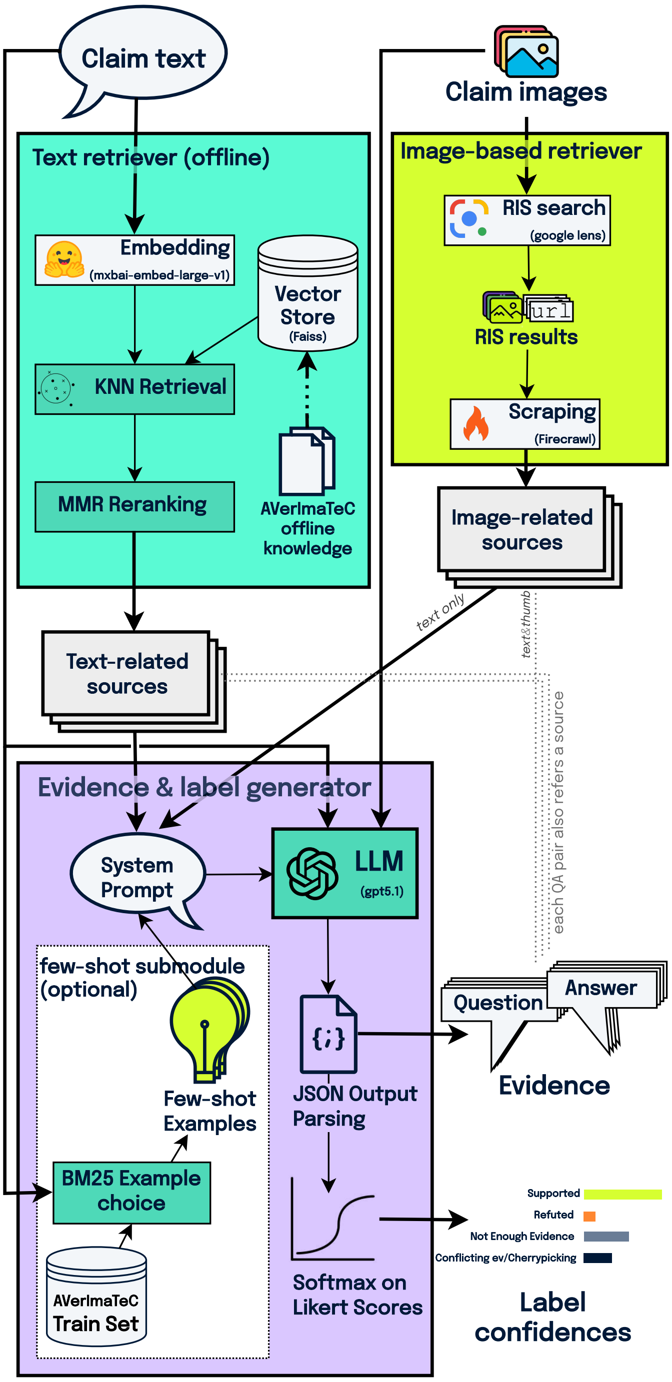

Figure 6.2: Our image-text fact-checking pipeline for the AVerImaTeC submission, extending Figure 6.1 with an image-based retrieval module. System is described in detail in Section 6.7.

Chapter 7
Agentic Fact-Checking

TODO: Write chapter introduction. Motivate the shift from pipeline-based (Chapters 4 and 6) to agentic architectures: dynamic planning, open-web retrieval, and LLM orchestration rather than fixed retrieval indices.

The systems described in Chapters 4 and 6 share a fundamental assumption: evidence is retrieved from a pre-built, fixed-scope corpus (a Wikipedia snapshot or a knowledge store constructed offline). While this closed-world design affords reproducibility and isolation from web noise, it limits the pipeline to facts that happen to be present in the corpus, and requires building and maintaining a language-specific retrieval index.

A complementary direction – increasingly prominent since the widespread availability of capable instruction-following LLMs– is to forgo a fixed corpus entirely, instead granting the model *tool access* to the open web and delegating both planning and evidence assessment to the model. This chapter describes `facticli`, our implementation of such an agentic pipeline, and presents **TODO:** benchmark results on AVeriTeC and ...(fill in once evaluation is complete).

7.1 Design Goals and Contrast with Prior Work

`facticli` was designed around several explicit goals:

1. **Open-world retrieval.** Rather than querying a pre-built index, the system issues targeted web searches at inference time. This removes corpus maintenance overhead and allows the system to access information published after any snapshot was taken – a property that is critical for the time-sensitive claims common in real-world fact-checking [Schlichtkrull et al., 2024b].

2. **Claim decomposition as a first-class step.** Inspired by the AVeriTeC [Schlichtkrull et al., 2024b] and QACG [Pan et al., 2021] evidence generation schemes, the system explicitly decomposes each claim into a set of *independent verification checks* before gathering evidence. This structured planning step mirrors the sub-question generation approach of AVeriTeC-style systems and allows individual checks to be executed concurrently.

3. **LLM synthesis.** Unlike the veracity modules based on NLI in prior chapters, the final verdict is produced by a judge LLM that reasons jointly over all collected evidence and assigns one of four veracity classes – `SUPPORTS`, `REFUTES`, `NEI`, or *Conflicting Evidence/Cherrypicking* – matching the AVeriTeC label scheme.

4. **Provider agnosticism.** The system is designed to run with any OpenAI-compatible inference endpoint (OpenAI, Gemini, or self-hosted models) and with either a hosted

web search tool or the Brave Search API, making it straightforward to swap model providers and measure their effect on performance.

7.2 Architecture

`facticli` follows a layered architecture with three explicit pipeline stages, each backed by a typed contract and a modular prompt skill. The full pipeline is depicted in Figure 7.1.

TODO: Insert figure: high-level diagram of the three-stage pipeline (Planner → parallel Researchers → Judge) with typed data contracts at each boundary.

Figure 7.1: Agentic fact-checking pipeline. Given an input claim, the *planner* decomposes it into independent verification checks. Each check is investigated concurrently by a *researcher* agent with web search tool access. The *judge* synthesises findings into a final veracity verdict and justification.

7.2.1 Planner Stage

The *planner* receives the claim text and produces an `InvestigationPlan`: a structured list of `VerificationCheck` objects, each containing a focused verification question, a rationale, and a set of targeted search queries. The planner is instructed to focus on verifiable atomic facts – dates, quantities, named entities, causal assertions – and to produce checks that can be executed by a separate agent without knowledge of each other's outcomes. A fallback single-check plan is generated automatically if planning fails.

7.2.2 Research Stage

Each `VerificationCheck` is executed concurrently by a separate *researcher* agent instance, bounded by a configurable semaphore to limit API concurrency. The researcher is equipped with a web search tool and instructed to prefer primary and high-credibility sources, to triangulate across multiple domains, and to treat time-sensitive claims as date-bound. It produces an `AspectFinding` with a discrete `EvidenceSignal` (*supports*, *refutes*, *mixed*, or *insufficient*), a confidence score, a grounded summary, and a source list with snippets. Failed checks are gracefully downgraded to *insufficient* findings rather than aborting the whole run.

7.2.3 Judge Stage

The *judge* receives the claim, the full investigation plan, and all aspect findings, and synthesises them into a `FactCheckReport`. The verdict is one of the four AVeriTeC-aligned classes; the judge assigns a calibrated `verdict_confidence` score, a tight evidence-grounded justification, and a deduplicated merged source list. The judge prompt specifies an explicit decision policy for handling *insufficient* findings: they are treated as absence of evidence rather than evidence of absence, and the verdict confidence is discounted proportionally.

7.2.4 Claim Extraction

A fourth, independent `ClaimExtractionStage` supports extracting atomic, decontextualized, check-worthy claims from arbitrary input text. Each extracted `CheckworthyClaim` includes a source fragment grounding it in the input, an identifier, and a rationale explaining why the claim is check-worthy. This component connects `facticli` to the claim extraction work of Chapter 5 and enables end-to-end pipelines from raw text to veracity verdicts.

7.3 Data Contracts

Each stage boundary is defined by typed Pydantic schemas:

- `InvestigationPlan` / `VerificationCheck` – planner output;
- `AspectFinding` / `SourceEvidence` – researcher output per check;
- `FactCheckReport` – final judge output.

Structured outputs constrain the LLM to valid schemas at each boundary, eliminating a major source of fragility in LLM pipelines and enabling downstream programmatic processing, evaluation, and logging of run artifacts.

7.4 Implementation Details

- **Inference backend.** All three stages are implemented via the OpenAI Agents SDK[1] using a unified OpenAI-compatible adapter; switching providers (OpenAI, Gemini) amounts to swapping API key and base URL only. **TODO:** Add formal citation for OpenAI Agents SDK once a citable reference is available.
- **Prompting strategy.** Each stage has a dedicated, minimal prompt skill (`plan.md`, `research.md`, `judge.md`) loaded from the repository, kept deliberately short and stage-specific to avoid cross-stage instruction leakage.
- **Parallelism.** Research checks run as `asyncio` tasks within a shared semaphore; the number of concurrent workers and the timeout per check are configurable at runtime.
- **AVeriTeC integration.** A dedicated submission script converts `facticli` output into the AVeriTeC evidence format (question–answer–URL triples), enabling direct evaluation via the AVeriTeC scorer [Schlichtkrull et al., 2024b].

7.5 Experimental Setup

TODO: Describe the experimental setup: which LLMs were evaluated (GPT-4.1, Gemini, ...), which search backends were used (OpenAI web search, Brave), max-checks and parallelism settings, and which benchmark splits were used.

TODO: Describe the AVeriTeC scorer and the AVeriTeC score metric, cross-referencing Chapter 6.

[1] `https://github.com/openai/openai-agents-python`

7.6 Results

TODO: Present main results table: AVeriTeC score on the AVeriTeC dev/test splits for each model–search-provider combination, compared with the baselines from Chapter 6 and the shared task leaderboard.

TODO: Analyse per-label performance and verdict confidence calibration.

TODO: If applicable, report results on additional benchmarks (FEVER, CTKFacts, ...).

7.7 Analysis

TODO: Discuss qualitative examples where agentic planning recovers evidence missed by fixed-corpus retrieval, and failure cases (time-sensitive claims, paywalled sources, hallucinated URLs).

TODO: Discuss effect of max-checks on performance vs. API cost.

TODO: Discuss source quality: do more checks or more search results lead to better evidence?

7.8 Conclusion

TODO: Summarise the chapter's contributions, limitations, and how this work connects to the broader dissertation narrative (from closed-world QACG-trained pipelines through AVeriTeC RAG to open-world LLM orchestration).

Chapter 8
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 SOTA 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* an LLM), 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 LLMs.

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.

8.1 Other contributions

Beyond the contributions reported in the preceding chapters, an important part of my doctoral work has also been the supervision of student research closely connected to automated fact-checking and Czech NLP more broadly. This section is intentionally only a short stub; the first contribution I want to record here is the master's thesis of Tomáš Mlynář.

A contribution I am particularly glad to have supported as a supervisor is Tomáš Mlynář's master's thesis, *Compute-constrained LLM adaptation to Czech language* [Mlynář, 2025]. The work itself and its findings are his contribution; my role here was supervisory. Mlynář 2025 studies how open-weight LLMs can be adapted to Czech under realistic compute constraints, with a systematic comparison of vocabulary adaptation, continued pretraining, instruction tuning, and Czech-English data mixing.

Among the most useful findings of [Mlynář, 2025] are that average-subword initialization speeds up convergence during vocabulary adaptation, that mixing Czech and English data mitigates catastrophic forgetting, and that none of the tested adaptation recipes consistently outperforms a strong Llama 3.1 8B Instruct baseline. The thesis also contributes reusable datasets, models, and evaluation resources, including Czech instruction-tuning and NLI collections and the Czech WildBench benchmark [Mlynář, 2025].

This is directly relevant to automated fact-checking. Modern fact-checking pipelines increasingly depend on instruction-following LLM components for claim extraction, question generation, evidence handling, summarization, and verdict support, and these components need strong Czech-language capabilities. Mlynář's work therefore clarifies what kinds of Czech LLM adaptation are already practically useful, what trade-offs remain unresolved, and which resources can be reused in future Czech fact-checking research Mlynář 2025. It was also recognized by FEL ČVUT among the awarded master's theses of 2025 [Fakulta elektrotechnická, České vysoké učení technické v Praze, 2025].

8.2 2026 SotA

Bibliography

Akhtar, M., Schlichtkrull, M., and Vlachos, A. (2024). Ev2R: Evaluating Evidence Retrieval in Automated Fact-Checking. Available at: https://arxiv.org/abs/2411.05375.

Allcott, H. and Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, 31(2):211–36. Available at: https://doi.org/10.1257/jep.31.2.211.

Aly, R., Christodoulopoulos, C., Cocarascu, O., Guo, Z., Mittal, A., Schlichtkrull, M., Thorne, J., and Vlachos, A., editors (2022). *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, Dublin, Ireland. Association for Computational Linguistics. Available at: https://aclanthology.org/2022.fever-1.0.

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)*.

Artstein, R. (2017). *Inter-annotator Agreement*, pages 297–313. Springer Netherlands, Dordrecht. Available at: https://doi.org/10.1007/978-94-024-0881-2_11.

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. Association for Computational Linguistics.

Barbaresi, A. (2020). Htmldate: A Python package to extract publication dates from web pages. *Journal of Open Source Software*, 5(51):2439.

Barrón-Cedeño, A., Alam, F., Caselli, T., Da San Martino, G., Elsayed, T., Galassi, A., Haouari, F., Ruggeri, F., Struss, J. M., Nandi, R. N., Cheema, G. S., Azizov, D., and Nakov, P. (2023). The clef-2023 checkthat! lab: Checkworthiness, subjectivity, political bias, factuality, and authority. In *Advances in Information Retrieval*, pages 506–517, Cham. Springer Nature Switzerland.

Barrón-Cedeño, A., Elsayed, T., Nakov, P., Martino, G. D. S., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., and Ali, Z. S. (2020). Overview of checkthat! 2020: Automatic identification and verification of claims in social media.

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. Available at: https://doi.org/10.1016/j.pdisas.2020.100119.

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, J. (1989). Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters. In *Advances in Neural Information Processing Systems*. Morgan-Kaufmann.

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. (2020). Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165. Available at: `https://arxiv.org/abs/2005.14165`.

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. Available at: `https://doi.org/10.1177/2056305119888654`.

Cao, R., Ding, Z., Guo, Z., Schlichtkrull, M., and Vlachos, A. (2025). AVerImaTeC: A Dataset for Automatic Verification of Image-Text Claims with Evidence from the Web. Available at: `https://arxiv.org/abs/2505.17978`.

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*, pages 335–336. Association for Computing Machinery. Available at: `https://doi.org/10.1145/290941.291025`.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879. Association for Computational Linguistics. Available at: `https://aclanthology.org/P17-1171`.

Chen, J., Kim, G., Sriram, A., Durrett, G., and Choi, E. (2023). Complex Claim Verification with Evidence Retrieved in the Wild. Available at: `https://arxiv.org/abs/2305.11859`.

Cheng, J., Dong, L., and Lapata, M. (2016). Long Short-Term Memory-Networks for Machine Reading. *CoRR*, abs/1601.06733.

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. Available at: `https://doi.org/10.1162/tacl_a_00377`.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzman, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics. Available at: `https://aclanthology.org/2020.acl-main.747`.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating Cross-lingual Sentence Representations. Available at: `https://aclanthology.org/D18-1269`.

Dammu, P. P. S., Naidu, H., Dewan, M., Kim, Y., Roosta, T., Chadha, A., and Shah, C. (2024). ClaimVer: Explainable claim-level verification and evidence attribution of text through knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13613–13627, Miami, Florida, USA. Association for Computational Linguistics. Available at: `https://aclanthology.org/2024.findings-emnlp.795`.

DeepL (2021). DeepL Translator. Available at: `https://www.deepl.com/translator` (accessed 2026-03-07).

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. Available at: https://arxiv.org/abs/2501.12948.

Déjean, H., Clinchant, S., and Formal, T. (2024). A Thorough Comparison of Cross-Encoders and LLMs for Reranking SPLADE. Available at: https://arxiv.org/abs/2403.10407.

Deng, Z., Schlichtkrull, M., and Vlachos, A. (2024). Document-level claim extraction and decontextualisation for fact-checking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11943–11954, Bangkok, Thailand. Association for Computational Linguistics. Available at: https://aclanthology.org/2024.acl-long.645.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. Available at: https://arxiv.org/abs/2305.14314.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/N19-1423.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. (2024). The Faiss library. *CoRR*, abs/2401.08281. Available at: https://arxiv.org/abs/2401.08281.

Drchal, J., Ullrich, H., Mlynář, T., and Moravec, V. (2024). Pipeline and dataset generation for automated fact-checking in almost any language. Available at: https://doi.org/10.1007/s00521-024-10113-5.

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. Available at: https://arxiv.org/abs/2407.21783.

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. Available at: https://arxiv.org/abs/2109.15118.

Fakulta elektrotechnická, České vysoké učení technické v Praze (2025). Nejlepší absolventi za rok 2025. Section "S oceněnou diplomovou prací". Available at: https://intranet.fel.cvut.cz/cz/education/ocenene-prace25.html (accessed 2026-03-08).

Fernandes Salvador, J. A. (2011). *Data analysis advances in marine science for fisheries management: Supervised classification applications*. PhD thesis, Universidad del País Vasco / Euskal Herriko Unibertsitatea. Available at: https://addi.ehu.es/handle/10810/8383.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382. Available at: https://doi.org/10.1037/h0031619.

Fu, J., Ng, S.-K., Jiang, Z., and Liu, P. (2024). GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics. Available at: https://aclanthology.org/2024.naacl-long.365.

Gažo, A. (2021). Algorithms for Document Retrieval in Czech Language Supporting Long Inputs. Available at: https://dspace.cvut.cz/handle/10467/97063 (accessed 2026-03-07).

Glockner, M., Hou, Y., and Gurevych, I. (2022). Missing Counter-Evidence Renders NLI Data Inadequate for Fact Verification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

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.

Google (2021). Cloud Translation - Google Cloud. Available at: `https://cloud.google.com/translate` (accessed 2026-03-07).

Guo, Z., Schlichtkrull, M., and Vlachos, A. (2022). A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206. Available at: `https://arxiv.org/abs/2108.11896`.

Gwet, K. L. (2010). *Handbook of Inter-rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters.* Advanced Analytics, LLC.

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. Association for Computational Linguistics.

Hasan, T., Bhattacharjee, A., Islam, M. S., Mubasshir, K., Li, Y.-F., Kang, Y.-B., Rahman, M. S., and Shahriyar, R. (2021). XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics. Available at: `https://aclanthology.org/2021.findings-acl.413`.

Hayes, A. and Krippendorff, K. (2007). Answering the Call for a Standard Reliability Measure for Coding Data. *Communication Methods and Measures*, 1:77–89. Available at: `https://doi.org/10.1080/19312450709336664`.

He, P., Gao, J., and Chen, W. (2023). DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. Available at: `https://arxiv.org/abs/2111.09543`.

He, P., Liu, X., Gao, J., and Chen, W. (2021). DEBeRTA: Decoding-enhanced BERT with disentangled attention.

Heilman, M., Cahill, A., Madnani, N., Lopez, M., Mulholland, M., and Tetreault, J. (2014). Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180, Baltimore, Maryland. Association for Computational Linguistics. Available at: `https://aclanthology.org/P14-2029`.

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.

Huguet Cabot, P.-L. and Navigli, R. (2021). REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics. Available at: `https://aclanthology.org/2021.findings-emnlp.204`.

Islam, M. A. and Magnani, E. (2021). Is this the end of the gold standard? A straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. Available at: `https://aclanthology.org/2021.emnlp-main.239`.

Jaradat, I., Zhang, H., and Li, C. (2024). On Context-aware Detection of Cherry-picking in News Reporting. Available at: `https://arxiv.org/abs/2401.05650`.

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. Available at: `https://doi.org/10.1145/3571730`.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b.

Johnson, J., Douze, M., and Jegou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547. Available at: `https://doi.org/10.1109/TBDATA.2019.2921572`.

Joshi, A., Kale, S., Chandel, S., and Pal, D. (2015). Likert Scale: Explored and Explained. *British Journal of Applied Science & Technology*, 7:396–403. Available at: `https://doi.org/10.9734/BJAST/2015/14975`.

Kann, K., Rothe, S., and Filippova, K. (2018). Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics. Available at: `https://aclanthology.org/K18-1031`.

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. Available at: `https://arxiv.org/abs/2112.01810`.

Koto, F., Baldwin, T., and Lau, J. H. (2020). FFCI: A Framework for Interpretable Automatic Evaluation of Summarization. Available at: `https://doi.org/10.48550/ARXIV.2011.13662`.

Krippendorff, K. (1970). Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement*, 30(1):61–70. Available at: `https://doi.org/10.1177/001316447003000105`.

Krippendorff, K. (2011). Computing Krippendorff's alpha-reliability. *Departmental Papers (ASC)*. Available at: `https://repository.upenn.edu/asc_papers/43`.

Krotil, M. (2022). Text Summarization Methods in Czech. Available at: `https://dspace.cvut.cz/handle/10467/101028` (accessed 2026-03-07).

Kuhn, T., Barbano, P. E., Nagy, M. L., and Krauthammer, M. (2013). *Broadening the Scope of Nanopublications*, pages 487–501. Springer Berlin Heidelberg, Berlin, Heidelberg. Available at: `https://doi.org/10.1007/978-3-642-38288-8_33`.

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*, pages 611–626. Association for Computing Machinery. Available at: `https://doi.org/10.1145/3600006.3613165`.

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. Available at: `https://arxiv.org/abs/2304.07327`.

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. Available at: `https://doi.org/10.1126/science.aao2998`.

Lee, S., Shakir, A., Koenig, D., and Lipp, J. (2024). Open Source Strikes Bread - New Fluffy Embeddings Model. Available at: `https://www.mixedbread.com/blog/mxbai-embed-large-v1` (accessed 2026-03-07).

Lehecka, J. and Svec, J. (2021). Comparison of Czech Transformers on Text Classification Tasks. In *Statistical Language and Speech Processing*, pages 27–37. Springer International Publishing.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, 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. Available at: `https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf`.

Li, X. and Li, J. (2024). AoE: Angle-optimized Embeddings for Semantic Textual Similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839. Association for Computational Linguistics.

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

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. (2022). Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. Available at: `https://arxiv.org/abs/2205.05638`.

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). Available at: `https://doi.org/10.1145/3560815`.

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. Available at: `https://arxiv.org/abs/2001.08210`.

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. Available at: `https://doi.org/10.1016/j.metrad.2023.100017`.

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, C. (2024). Multi-hop Evidence Pursuit Meets the Web: Team Papelo at FEVER 2024. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 27–36. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/2024.fever-1.2`.

Mao, Y., Zhong, M., and Han, J. (2022). CiteSum: Citation text-guided scientific extreme summarization and domain adaptation with limited supervision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10922–10935, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. Available at: `https://aclanthology.org/2022.emnlp-main.750`.

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. Available at: `https://arxiv.org/abs/2203.11147`.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Available at: `https://arxiv.org/abs/1301.3781`.

Mlynář, T. (2023). Automated Fact Checking Based on Czech Wikipedia. Available at: `https://dspace.cvut.cz/handle/10467/109219` (accessed 2026-03-07).

Mlynář, T. (2025). Compute-constrained LLM adaptation to czech language. Master's thesis, Czech Technical University in Prague, Faculty of Electrical Engineering. Available at: `http://hdl.handle.net/10467/123587`.

Mroczkowski, R., Rybak, P., Wroblewska, 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. Association for Computational Linguistics.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/2023.eacl-main.148`.

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. Available at: `https://doi.org/10.1007/978-3-030-85251-1_19`.

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. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/D18-1206`.

Nie, Y., Chen, H., and Bansal, M. (2019). Combining Fact Extraction and Verification with Neural Semantic Matching Networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901. Association for Computational Linguistics. Available at: `https://aclanthology.org/2020.acl-main.441`.

NLP-Progress (2023). On Summarization. Available at: `https://nlpprogress.com/english/summarization.html` (accessed 2026-03-07).

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. Linköping University Electronic Press, Sweden.

OpenAI (2023a). GPT-4 Technical Report. Available at: `https://cdn.openai.com/papers/gpt-4.pdf`.

OpenAI (2023b). What are tokens and how to count them? Available at: `https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them` (accessed 2026-03-07).

OpenAI (2024). Introducing structured outputs in the API. Available at: `https://openai.com/index/introducing-structured-outputs-in-the-api/` (accessed 2026-03-07).

Pan, L., Chen, W., Xiong, W., Kan, M.-Y., and Wang, W. Y. (2021). Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics. Available at: `https://aclanthology.org/2021.acl-short.61`.

Patel, D. and Ahmad, A. (2023). Google "We Have No Moat, And Neither Does OpenAI". Available at: `https://semianalysis.com/2023/05/04/google-we-have-no-moat-and-neither/` (accessed 2026-03-07).

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, 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. Association for Computational Linguistics. Available at: `https://doi.org/10.3115/v1/D14-1162`.

Pikuliak, M., Grivalský, Š., Konôpka, M., Blšták, M., Tamajka, M., Bachratý, V., Šimko, M., Balážik, P., Trnka, M., and Uhlárik, F. (2021). SlovakBERT: Slovak Masked Language Model. Available at: `https://arxiv.org/abs/2109.15254`.

Pomerlau, D. and Rao, D. (2017). Fake News Challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. Available at: `http://www.fakenewschallenge.org/` (accessed 2026-03-07).

Popel, M., Tomkova, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar, O., and Zabokrtsky, Z. (2020). Transforming machine translation: A deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15. Available at: `https://doi.org/10.1038/s41467-020-18073-9`.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. Available at: `https://jmlr.org/papers/v21/20-074.html`.

Raheja, V., Kumar, D., Koo, R., and Kang, D. (2023). CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics. Available at: `https://aclanthology.org/2023.findings-emnlp.350`.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/D16-1264`.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. Available at: `https://aclanthology.org/D19-1410`.

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)*.

Rufibach, K. (2010). Use of Brier score to assess binary predictions. *Journal of Clinical Epidemiology*, 63(8):938–939. Available at: `https://doi.org/10.1016/j.jclinepi.2009.11.009`.

Rýpar, M. (2021). Methods of Document Retrieval for Fact-Checking. Available at: `https://dspace.cvut.cz/handle/10467/95315` (accessed 2026-03-07).

Sadvilkar, N. and Neumann, M. (2020). PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics. Available at: `https://aclanthology.org/2020.nlposs-1.15`.

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 *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/2024.fever-1.1`.

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*. Curran Associates Inc.

Schuster, T., Fisch, A., and Barzilay, R. (2021). Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. *CoRR*, abs/2103.08541.

Sebastian, G. (2023). Exploring Ethical Implications of ChatGPT and Other AI Chatbots and Regulation of Disinformation Propagation. Available at: `https://doi.org/10.2139/ssrn.4461801`.

Semin, D. (2023). Multitask Learning for NLP Classifiers. Available at: `https://dspace.cvut.cz/handle/10467/109243` (accessed 2026-03-07).

Sheikhi, G., Touileb, S., and Khan, S. (2023). Automated claim detection for fact-checking: A case study using Norwegian pre-trained language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 1–9, Tórshavn, Faroe Islands. University of Tartu Library. Available at: `https://aclanthology.org/2023.nodalida-1.1`.

Sido, J., Pražák, O., Přibáň, P., Pašek, J., Seják, M., and Konopík, M. (2021). Czert – Czech BERT-like Model for Language Representation. Available at: `https://doi.org/10.26615/978-954-452-072-4_149`.

Stănescu, G. (2022). Ukraine conflict: The challenge of informational war. *SOCIAL SCIENCES AND EDUCATION RESEARCH REVIEW*, 9(1):146–148. Available at: `https://doi.org/10.5281/zenodo.6795674`.

Stefanik, M., Kadlcík, 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. Association for Computational Linguistics.

STEM (2021). Mýtům a konspiracím o COVID-19 věří více než třetina české internetové populace | STEM.cz. Available at: `https://www.stem.cz/mytum-a-konspiracim-o-covid-19-veri-vice-nez-tretina-ceske-internetove-populace/` (accessed 2026-03-07).

Straka, M., Náplava, J., Straková, J., and Samuel, D. (2021). RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. *Lecture Notes in Computer Science*, pages 197–209. Available at: `https://doi.org/10.1007/978-3-030-83527-9_17`.

Strakova, J., Straka, M., and Hajic, J. (2019). Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331. Association for Computational Linguistics. Available at: `https://doi.org/10.18653/v1/P19-1527`.

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. European Language Resources Association.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks.

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? Available at: `https://arxiv.org/abs/2309.08963`.

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. Available at: `https://crfm.stanford.edu/2023/03/13/alpaca` (accessed 2026-03-07).

Thorne, J. and Vlachos, A. (2021). Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. Association for Computational Linguistics. Available at: https://aclanthology.org/2021.acl-long.256.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018a). FEVER: A Large-scale Dataset for Fact Extraction and VERification. In *NAACL-HLT*.

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A. (2018b). The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

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. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/D19-6601.

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. Available at: https://arxiv.org/abs/2302.13971.

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. Available at: https://arxiv.org/abs/2307.09288.

Ullrich, H. (2021). Dataset for Automated Fact Checking in Czech Language. Available at: https://dspace.cvut.cz/handle/10467/95430 (accessed 2026-03-07).

Ullrich, H. and Drchal, J. (2025). AIC CTU System for the 2nd AVeriTeC Shared Task. In *Proceedings of the 2nd Workshop on Automated Fact Verification*.

Ullrich, H., Drchal, J., Rypar, M., Vincourova, H., and Moravec, V. (2023). CsFEVER and CTKFacts: Acquiring Czech data for fact verification. *Language Resources and Evaluation*. Available at: https://doi.org/10.1007/s10579-023-09654-3.

Ullrich, H., Mlynar, T., and Drchal, J. (2024). AIC CTU system at AVeriTeC: Re-framing automated fact-checking as a simple RAG task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 137–150. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/2024.fever-1.16.

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: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. Available at: https://lmsys.org/blog/2023-03-30-vicuna/ (accessed 2026-03-07).

Vijayakumar, A., Cogswell, M., Selvaraju, R., Sun, Q., Lee, S., Crandall, D., and Batra, D. (2018). Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). Available at: https://ojs.aaai.org/index.php/AAAI/article/view/12340.

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. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/2020.acl-main.450.

Wardle, C. and Derakhshan, H. (2017). *Information Disorder: Toward an Interdisciplinary Framework for Research and Policymaking*. Council of Europe. Available at: https://shorensteincenter.org/resource/information-disorder-framework-for-research-and-policymaking/ (accessed 2026-03-07).

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*. Curran Associates Inc.

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, 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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/2020.emnlp-demos.6.

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. Available at: https://aclanthology.org/2022.acl-long.175.

Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics. Available at: https://aclanthology.org/2020.emnlp-main.523.

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. Available at: https://arxiv.org/abs/2505.09388.

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. Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/2021.emnlp-main.611.

Zha, Y., Yang, Y., Li, R., and Hu, Z. (2023). AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics. Available at: https://aclanthology.org/2023.acl-long.634.

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, 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. Available at: https://arxiv.org/abs/2303.18223.

Appendix A
Acronyms

ČTK Czech Press Agency

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

FEVER Fact Extraction and VERification

GPT Generative Pre-trained Transformer

GPU Graphics Processing Unit

IR Information Retrieval

JSON JavaScript Object Notation

LLM Large Language Model

MMR Maximal Marginal Relevance

NER Named Entity Recognition

NLI Natural Language Inference

NLP Natural Language Processing

QA Question Answering

RAG Retrieval-Augmented Generation

RIS Reverse Image Search

RTE Recognizing Textual Entailment

SOTA State of the Art

XSum Extreme Summarization

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 Multimodal system prompt

```
You are a professional fact checker of image-text claims, 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. The claim consists of a textual statement and
  ↪ {image_count} images associated with the claim. The claim was made by {author} on {date} via
  ↪ {medium}. 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 the sources listed below. The first {k} sources was retrieved using
  ↪ textual search and the rest was retrieved using reverse image search (google lens). The
  ↪ sources are numbered - sources 1 through {k} are related to the claim text,  sources 11-19
  ↪ were retrieved for the first user image, 21-29 to the second etc. You may therefore assume
  ↪ that each of the image-based sources was published alongside a picture similar to the
  ↪ respective user image.
---
## Source ID: 1 [url]
[context before]
[page content]
[context after]
...
---
## Image Source ID: 11 (related to user image 1, Title : [title], date:[page_date], url: [url],
  ↪ image url: [img_url])
[content]
...
---
## 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>"},...   ],
 "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>",
  "verdict_justification": "<A brief justification of the veracity verdict>"
}
```

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.7: Our fact-checking system prompt to be used with Multimodal LLM, feeding the AVerImaTeC claim text and images into its multimodal user message. Three dots represent omitted repeating parts of the prompt. Adapted for multimodal scenario from Ullrich and Drchal 2025.

B.4 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.8: Qwen3 thinking tokens example.