Exploring Semantic Filtering Heuristics For Efficient Claim Verification

Max Upravitelev[1], Premtim Sahitaj[1], Arthur Hilbert[1], Veronika Solopova[1], Jing Yang[1,3], Nils Feldhus[1,3], Tatiana Anikina[2], Simon Ostermann[2,4] and Vera Schmitt[1,2,3,4]
[1]Technische Universität Berlin
[2]German Research Center for Artificial Intelligence (DFKI)
[3]BIFOLD – Berlin Institute for the Foundations of Learning and Data
[4]Centre for European Research in Trusted AI (CERTAIN)
Correspondence: max.upravitelev@tu-berlin.de

Abstract

Given the limited computational and financial resources of news agencies, real-life usage of fact-checking systems requires fast response times. For this reason, our submission to the FEVER-8 claim verification shared task focuses on optimizing the efficiency of such pipelines built around subtasks such as evidence retrieval and veracity prediction. We propose the Semantic Filtering for Efficient Fact Checking (SFEFC) strategy, which is inspired by the FEVER-8 baseline and designed with the goal of reducing the number of LLM calls and other computationally expensive subroutines. Furthermore, we explore the reuse of cosine similarities initially calculated within a dense retrieval step to retrieve the top 10 most relevant evidence sentence sets. We use these sets for semantic filtering methods based on similarity scores and create filters for particularly hard classification labels "Not Enough Information" and "Conflicting Evidence/Cherrypicking" by identifying thresholds for potentially relevant information and the semantic variance within these sets. Compared to the parallelized FEVER-8 baseline, which takes 33.88 seconds on average to process a claim according to the FEVER-8 shared task leaderboard, our non-parallelized system remains competitive in regard to AVeriTeC retrieval scores while reducing the runtime to 7.01 seconds, achieving the fastest average runtime per claim.

1 Introduction

Building systems for claim verification poses a significant challenge and is typically evaluated with accuracy-related metrics. At the same time, the efficiency of systems generally proposed within natural language processing (NLP) research is becoming another major aspect of system design (Treviso et al., 2023), motivated by sustainability and efforts in the domain of green NLP (Strubell et al., 2019). The field is experiencing a gradual conceptual shift towards smaller, more efficient models, as evidenced by the proliferation of smaller open-source transformer models (e.g., Gemma 3 (Team et al., 2025), Llama 3.2 (Grattafiori et al., 2024), or Phi 4 (Abdin et al., 2024)). Commercial proprietary models follow the trend, with GPT-4o (OpenAI et al., 2024) being the default flagship OpenAI model, while also being twice as fast and 50% more cost-effective than the larger GPT-4 Turbo model (OpenAI, 2023). DeepSeek (DeepSeek-AI et al., 2024) released a smaller model that matches or exceeds the performance of GPT models in various tasks, while requiring significantly less computational power for inference.

In parallel, embedding models used for semantic similarity search are undergoing a similar transformation. Open-source models such as E5-small (Wang et al., 2022) and MiniLM (Wang et al., 2020) demonstrate that compact architectures can achieve retrieval performance competitive with LLMs while significantly reducing inference costs. More recently, models such as GTE (Li et al., 2023a) and BGE (Liu et al., 2023) have gained attention for offering strong performance on a variety of retrieval tasks with relatively lightweight configurations. At the commercial level, OpenAI's text-embedding-3-small model (OpenAI, 2024) achieves strong semantic performance at a fraction of the cost and latency of earlier embedding APIs. This collective shift reflects a growing emphasis on deployment efficiency, edge compatibility, and environmentally conscious model design, without compromising retrieval accuracy.

These transformations have a direct impact on Retrieval-Augmented Generation (RAG) systems, particularly in scenarios where cost-efficiency is critical. RAG systems typically involve a similarity search followed by an LLM re-assessment of the highest-ranking candidate data reference points. Efficiency-optimized RAG systems are particularly

Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER), pages 229–237

valuable in fact-checking applications, where news agencies often operate with limited computational resources, as well as understaffed and underfunded fact-checking units (Graves, 2018).

Taking these trends into account, we propose a system inspired by the FEVER-8 baseline with the main goal of reducing its runtime per claim while retaining comparable performance. To achieve this, we explore possibilities of reducing computationally expensive subroutines like generating texts with LLMs and the re-usage of calculated cosine similarities for the application of semantic filters for veracity prediction. Our main contributions are[1]:

- Introducing a pipeline designed for efficiency-aware claim verification that remains competitive with the FEVER-8 baseline in terms of retrieval scores, while reducing the average runtime per claim from 33.88s to 7.01s according to the FEVER-8 Leaderboard[2].
- Exploring the application of semantic filters to predict the veracity labels "Not Enough Information" and "Conflicting Evidence/Cherrypicking".

2 Related Work

Efficient fact-checking Tang et al. (2024) used GPT-4-generated training data to train small language models (770M parameters) for fact verification and showed their models achieving performance in closed-book and document-based settings on par with GPT-4. Xie et al. (2025) integrated interactive retrieval and verification and reduced costs for both parts, particularly for GPT-4o-mini on factuality benchmarks. The approach also enables LLMs to leverage their internal knowledge for judgments instead of always relying on external evidence retrieval.

These contributions achieved notable results in regard to building efficient systems and were evaluated on different benchmarks, a direction we also aim to contribute to with a system specifically tailored for the AVeriTeC dataset.

Semantic filtering Gupta et al. (2023) employed similarity metrics to perform semantic matching. In particular, they find mappings between scientific evidence in publications and paraphrased findings in health news articles using most similar paragraphs as evidence in the context of fake news detection.

The identification of thresholds for classification tasks based on cosine similarity was explored in works such as Pilehvar and Camacho-Collados (2019) and Zhou et al. (2022). Here, optimal thresholds were tuned by incrementing values stepwise while iterating over training sets with promising results with regard to performance and efficiency. Both works identified thresholds for binary classifiers to evaluate different word embedding models, with the goal of measuring the cosine distances between word pairs in different contexts. However, to our knowledge, this technique has not yet been examined in the context of veracity prediction.

3 Methodology

The AVeriTeC (Automated Verification of Textual Claims) dataset (Schlichtkrull et al., 2023) contains 4568 fact-checked, real-word claims. The data set enables the assessment of claim verification systems that retrieve evidence from the open web. AVeriTeC provides a training, development (dev) and test set. It is accompanied by a knowledge store with scraped texts from websites related to potential search queries to verify a claim. All claims are classified into four categories of verdicts:

- "Supported" (SUP)
- "Refuted" (REF)
- "Not Enough Evidence" (NEI)
- "Conflicting Evidence/Cherrypicking" (CoC).

Efficiency-optimized pipeline design One of the goals of the FEVER-8 shared task is the exploration of the usage of Open Source (OS) models while emphasizing the efficiency of the proposed systems by capping the maximum runtime at one minute per claim. Hence, a baseline was released based on an optimized version of HerO (Yoon et al., 2024), which was the highest scoring system from the FEVER-7 shared task (Schlichtkrull et al., 2024) that was built upon OS models. We designed our system inspired by this baseline and with the goal in mind of building a pipeline that reduces the amount of LLM calls and thus reduces the overall runtime. The resulting SFEFC pipeline relies on only two LLM calls within the following steps (which are also illustrated in Figure 1).

[1]Our code is available at: https://github.com/XplaiNLP/SFEFC-FEVER-8-Shared-Task

[2]https://fever.ai/task.html

1. Generate a question based on the claim by prompting an LLM
    a) Get a claim from the dataset, such as "In a letter to Steve Jobs, Sean Connery refused to appear in an apple commercial"
    b) Generate a corresponding question, such as "Is there any documented evidence or credible source that confirms Sean Connery wrote a letter to Steve Jobs refusing to appear in an Apple commercial?"
2. Retrieve relevant evidence based on hybrid search:
    a) Concatenate all individual sentences from the AVeriTeC knowledge store sequentially to sets of 4
    b) Sparse retrieval: Return top 1500 sets of sentence sets from 2.a via BM25
    c) Dense retrieval: Return the top 10 sentence sets based on the cosine similarity of the results from 2.b
3. Predict a verdict label for NEI and CoC via semantic filtering using cosine similarities from 2.c
4. If a semantic filter can be applied following Algorithm 1, the corresponding label is used as the final verdict
5. If not, an LLM prompt (included in the appendix of this paper) is constructed from the retrieved evidence, the claim and the system prompt with instructions to choose between SUP or REF, effectively reducing the classification labels to a binary choice in this step
6. The LLM prediction is used as the final verdict

The hybrid search step is similar to the same step within the FEVER-8/HerO baseline, but with one key change that we implemented to improve the runtime: While the baseline retrieved sentences one by one, we sequentially concatenated the sentences related to each claim in the AVeriTeC knowledge store into sets of 4 before retrieving these sets with BM25. The amount of 4 was chosen due to observations within preliminary testing, where we observed an increase in the old AVeriTeC score while incrementing the value until the amount of 4.

This concatenation strategy is similar to chunking strategies within RAG pipelines, where the size of chunks is determined by trade-offs, like preserving semantic information while keeping it precise enough for query-based retrieval. As a consequence, by concatenating the sentences in sets of 4 we also reduced the total number of retrieval candidates from the knowledge store by a factor of 4, allowing us to also reduce the number of top k results retrieved by BM25 from 5000 in the baseline to 1500. This, in turn, led to a decrease in the amount of embeddings needed to be created to retrieve the top 10 candidates based on cosine similarity. The actual values of the set size and the top *k* from BM25 were determined within preliminary tests by evaluating this pipeline on the AVeriTeC dev set with different values.

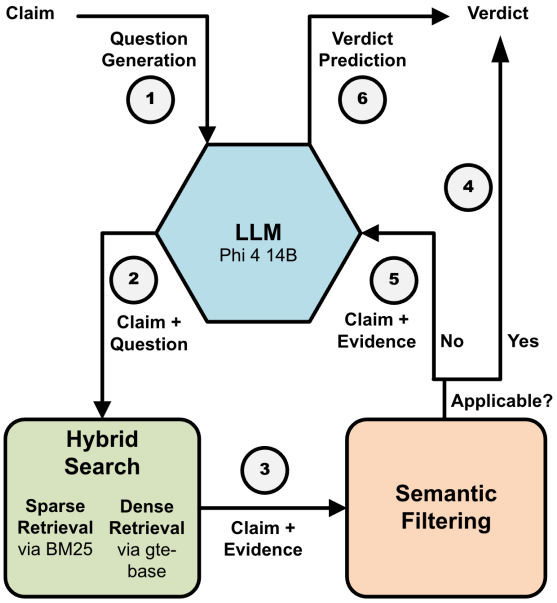

Figure 1: Architecture of the proposed system

Semantic filtering The proposed semantic filtering method aims to classify the veracity verdicts by assessing the cosine similarities between the query and the top 10 retrieved evidence. Algorithm 1 documents our Semantic Filtering for Efficient Fact Checking (SFEFC) approach, which was motivated by further lowering the runtime by reusing the cosine similarities during the 2.c step. Here, the reduction in runtime was achieved by removing an additional LLM call and using already computed similarity values to classify veracity labels, provided that a semantic filter was applicable.

The filtering strategy for the NEI class assumes a threshold of cosine similarity below which the retrieved information can be classified as not relevant.

The idea of filtering for *semantic variance* in regard to the CoC class follows the intuition that the variance within the set of cosine similarities between the query and the retrieved top 10 results should be higher in this class compared to others, since it should include evidence which both supports and refutes the claim. We calculate the semantic variance with

$$\mathrm{Var}(a) = \frac{1}{N}\sum_{i=1}^{N}|a_i - \bar{a}|^2$$

, where a_i are the elements in the input array a (consisting of the top 10 cosine similarities from the 2.c step), $\bar{a}$ is the mean of all elements and N is the total amount of all elements.

For identifying the thresholds, we implemented a similar routine to threshold tuning as Pilehvar and Camacho-Collados (2019) and Zhou et al. (2022): We incrementally increased the thresholds while iterating over our final prediction files for the dev set containing the retrieved evidence, updated the NEI and CoC labels if they met certain thresholds, and scored the overall predictions using the official AVeriTeC scorer which considered the metrics based on Hungarian METEOR (Schlichtkrull et al., 2023).

During the rise of OS LLMs, several inference engines with different trade-offs in regard to factors such as performance metrics and target hardware were released in recent years (Park et al., 2025). We chose to use llama.cpp[3] and its GGUF-quantization format for our LLM calls. Although lacking some capabilities of other engines, such as running inference on multiple nodes, llama.cpp was designed with the goal in mind of deploying LLMs on consumer hardware, making it a suitable choice that aligns with our motivation.

Name	Split	Hungarian METEOR	Ev2R
FEVER-8/HerO	dev	0.554	0.296
	test	0.497	0.2023
SFEFC-phi4	dev	0.572	0.266
	test	0.494	0.2047

Table 1: FEVER-8/HerO (baseline) and SFEFC-phi4 (our) results on the AVeriTeC dataset

[3] https://github.com/ggml-org/llama.cpp

Algorithm 1 Semantic Filtering with Heuristics

```
function GETLABEL(cos_sims)
    t_nei ← 0.82
    t_conf ← 0.0007
    pred ← None
    if cos_sims[0] < t_nei then
        pred ← "Not Enough Evidence"
    end if
    if Var(cos_sims) > t_conf then
        pred ← "Conflicting Evidence/ Cherrypicking"
    end if
    return pred
end function

function APPLYFILTER(claim, evs)
    cos_sims = [ ]
    for ev in evs do
        sim ← COSSIM(claim, ev)
        cos_sims.append(sim)
    end for
    label ← GETLABEL(cos_sims)
    return label
end function

function PREDICTLABEL(claim, evs)
    label ← APPLYFILTER(claim, evs)
    if label = None then
        label ← LLMPRED(claim, evs)
    end if
end function
```

4 Evaluation

Table 1 documents our results on the official FEVER-8 shared task leaderboards with regard to the dev set[4] and the test set[5]. Ev2R refers to the new AVeriTeC score Ev2R recall (Akhtar et al., 2024), a new metric for LLM-based evaluation of retrieval tasks. Unlike the old AVeriTeC score, which focused on a lexical metric, the new score considers the semantic meaning of the retrieved evidence. Both metrics consider correctly predicted verdicts and evidence retrieved for their prediction.

Experimental setup We ran different configurations of the baseline and our own system on the

[4] https://huggingface.co/spaces/fever/AVeriTeC Fever8Dev

[5] https://fever.ai/task.html

Block	Name	Old S	New S	s/claim	SUP	REF	NEI	CoC
1	FEVER-8/HerO	0.534	0.280	10.34	0.639	0.799	0.133	0.075
	FEVER-8/HerO-phi4-14B	0.522	0.256	13.13	0.623	0.783	0.103	0.000
2	SFEFC-phi4-14B	0.572	0.296	04.83	0.645	0.806	0.046	0.063
	SFEFC-phi4-14B-no-concat	0.452	0.248	05.33	0.525	0.784	0.000	0.033
3	SFEFC-phi4mini-3B	0.472	0.224	**04.28**	0.517	0.707	0.000	0.033
	SFEFC-llama3.2-3B	0.436	0.230	04.61	0.437	0.706	0.046	0.065
	SFEFC-gemma-3-27-it-qat	0.502	0.276	05.21	0.647	0.784	0.051	0.035
4	SFEFC-phi4-14B-all-classes	0.394	0.254	05.39	0.646	0.516	**0.163**	**0.280**
	SFEFC-phi4-14B-binary	**0.608**	**0.318**	04.68	**0.66**	**0.836**	0.000	0.000
	SFEFC-phi4-14B-varifocal	0.540	0.286	17.68	0.654	0.815	0.000	0.065

Table 2: Comparison of AVeriTeC scores (Old S: Hungarian METEOR score, New S: Ev2R Recall score), runtimes and Veracity F1 scores on the labels Supported (SUP), Refuted (REF), Not Enough Information (NEI) and Conflicting Evidence/Cherrypicking (CoC)

same machine with an NVIDIA H100 80GB GPU[6]. Both generative tasks, question generation and veracity prediction, were handled by the same LLM in each case. For better comparison, all configurations in Table 2 including HerO were evaluated with the same gte-base embedding model (Li et al., 2023b). Table 2 collects our evaluation results:

- Block 1 presents the FEVER-8/HerO-baseline results when run on our infrastructure
- Block 2 documents the results of our final submission. It also includes the results of a configuration with our concatenation strategy ablated.
- Block 3 collects the results of configurations were the LLM was replaced with other variants
- Block 4 shows strategies deviating from our main configuration

Runtime and accuracy-related metrics For the evaluation, our goal is to assess whether we could remain competitive with the baseline while improving the runtime per claim. As the results documented in Table 2 indicate, we were able to cut the runtime compared to the FEVER-8/HerO-baseline by more than 1/2 with most of our configurations (e.g., from 10.34s to 04.83 with our main configuration SFEFC-phi4-14B), while staying competitive both in the dev and the test set on the respective accuracy metrics (Table 1). Furthermore, the runtime of SFEFC-phi4-14B (Block 2) can be optimized in a way similar to the batching and parallelization strategies of the optimized FEVER-8/HerO-Pipeline for the FEVER-8 shared task. These strategies can also be explored with our proposed system to further reduce the processing time per claim in future work.

We experimented with multilingual-e5 (Wang et al., 2024), BGE-M3 (Chen et al., 2024), and gte-base (Li et al., 2023b), where gte-base yielded the best results in terms of runtime and accuracy.

We used Q6_0 GGUF[7] quantization variants in most configurations. Quantizing an LLM to 6-bit precision from 16-bit, as in the case of Phi 4, greatly reduces the runtime by lowering the requirements for in-memory operations.

As expected, the results indicate that runtime increases with the parameter size of the model, reflecting the higher computational cost of larger LLMs. Similarly, accuracy-related scores tend to decrease as the model size is reduced, illustrating a typical trade-off between efficiency and performance. Here, the Phi-4-14B (Abdin et al., 2024) model yielded the best results, while the related Phi-4-mini-instruct variant (Microsoft et al., 2025) with 3.8B parameters performed worse, as well as llama3.2-3B (Grattafiori et al., 2024) with 3.2 parameters. During our preliminary tests, we noticed better results when working with the Phi 4 model.

[6]The runtime values differ from the values on the FEVER-8 Leaderboard, where all systems were run on NVIDIA A10G GPUs with 23GB VRAM

[7]A quantization format developed by llama.cpp: https://github.com/ggml-org/ggml/blob/master/docs/gguf.md

For example, we evaluated our system in configuration with Gemma 3 (Team et al., 2025) with 27B parameters in its Quantization Aware Training (QAT) format, but the results were subpar compared to the Phi 4 variant. However, when running the FEVER-8/HerO-baseline while replacing the older Meta-Llama-3.1-8B-Instruct model with Phi 4 (on all tasks except veracity prediction), the retrieval score dropped. Thus, it cannot be generalized that the Phi 4 performs better in all cases. Furthermore, we evaluated the SFEFC-phi4-14B-all-classes to analyze how Phi 4 would perform if all four veracity labels were predicted by the model, with this configuration yielding the lowest scores (except, surprisingly, on the NEI and CoC classes).

Ablation Another goal of our evaluation documented in Block 2 of Table 2 is the comparison of different configurations to analyze their influence. To observe how our concatenation strategy influences the results, we evaluated the configuration SFEFC-phi4-14B-no-concat, which follows the same strategy as the FEVER-8/HerO-baseline (retrieving top 5000 individual sentences instead of 1500 concatenated sets of 4). As the results show, removing our strategy led to a drop in the veracity score (from 0.572 to 0.452) and an increase in runtime (from 4.84 to 5.33 seconds). This matches our assumption that retrieval performance increases when potential candidates contain more semantic information while decreasing runtime, since the total amount of potential candidates is also reduced by a factor of 4.

Further experiments We experimented with different strategies to further move beyond the results of SFEFC-phi4-14B, such as fine-tuning Phi 4 for question generation, fine-tuning BERT-based classifiers for verdict prediction, or different prompting strategies, but without success. As an example, we include our SFEFC-phi4-14B-varifocal variant, which was inspired by (Ousidhoum et al., 2022). Here, we prompted Phi 4 to generate 3 varifocal questions, parsed them from the output, used the generated questions as queries, and merged the results within the set of 10 retrieved question/evidence pairs. Although being a more sophisticated prompting strategy than generating a single question, the score did not improve against the SFEFC-phi4-14B configuration.

Error analysis When considering the results of our SFEFC-phi4-binary and SFEFC-phi4 configurations in Table 2, it is noticeable that while semantic filtering successfully labels some of the NEI and CoC cases, the veracity scores for SUP and REF drop by around 0.02-0.03 points. Thus, the actual number of correctly labeled NEI and CoC cases needs to be further examined. Table 3 illustrates that while predicting veracity classes with Semantic Filters is generally possible, mislabeled verdicts outweigh by a margin. When comparing the wrong predictions with their actual target labels, out of the 7 times NEI was mislabeled, 5 cases were actually REF and 2 SUP cases. In regard to CoC, out of the 26 wrong predictions, 17 should have been REF, 6 SUP and 1 NEI. While these results could point to a higher possibility of mislabeling predictions when the actual target label is REF, the ratios also roughly represent the class balance of the dev set (which includes 0.61% REF, 0.24% SUP, 0.08 % NEI and %0.07 CoC labeled claims). Thus, we conclude that while static thresholds can indeed be applied to predict correct veracity labels, the filtering strategy proposed in this paper can not be generalized to most cases, regardless of the actual target label.

Class	Total Actual	Total Predicted	TP	FP
NEI	35	8	1	7
CoC	38	26	2	24

Table 3: Examination of Not Enough Information (NEI) and Conflicting Evidence/Cherrypicking (CoC) cases. TP is short for True Positives/correctly predicted. FP is short for False Positives/incorrectly predicted.

5 Discussion and Future Work

With the proposed system, we successfully achieved our goal of remaining competitive with the FEVER-8 baseline in terms of performance, while significantly reducing runtime and taking a step closer toward real-world applicability in end-user-facing fact-checking systems. A key efficiency gain was achieved by reducing LLM calls and delegating both question generation and part of the veracity prediction to a single Phi-4 model, which, as our results show, performs well on both tasks. Another key approach involved concatenating sentences from the knowledge store based on chunking strategies, which significantly reduced the number of sentence embeddings required during each dense retrieval step. Both strategies can be further opti-

mized in future work to decrease the runtime per claim. Another promising direction is to further optimize runtime through the application of parallelization techniques, similar to those used in the FEVER-8/HerO baseline.

While our semantic filtering heuristics were able to correctly identify some veracity classes, they often resulted in incorrect labels overall. This suggests that, although the approach holds promise, it requires further refinement. Improvements could involve enhancing the current heuristic techniques or replacing them with learnable components within classification models. In particular, the use of *semantic variance* for predicting the challenging CoC class appears to be a promising direction. This method could evolve beyond fixed thresholding towards more flexible classifiers that leverage deeper features of the retrieved evidence embeddings.

Limitations

While we were able to reduce the runtime by around half when compared to the FEVER-8/AVeriTeC baseline, it still remains at 04.28 seconds per claim in our fastest configuration – a value that can be considered too slow for real-life settings, especially when taking into account that its achievement is limited to the system being run on a high-end GPU (NVIDIA H100).

Thus, further improvement is needed towards the deployment outside of laboratory settings and on lower-end devices, where scalability issues, latency requirements, and different deployment options need to be taken into account. There are several paths discussed going towards this goal in the current literature. For example, the runtime of the dense retrieval steps could be improved by the binarization of the embedding vectors, as discussed in Gan et al. (2023).

The performance of our system and the thresholds of our proposed semantic filtering methods are limited to the AVeriTeC dataset. As discussed in our error analysis section, these heuristics showcase the general possibility to filter out specific labels based on thresholds but need further refinement due to a large amount of false positives. This point is underscored by the need of assessment of the methods on other data beyond the AVeriTeC dataset.

Furthermore, the thresholds we identified are limited to the gte-base embedding model. The implementation of other members of this model class can result in different results due to differing ranges in cosine similarity scores.

Acknowledgments

The work on this paper is performed in the scope of the projects “VeraExtract” (01IS24066) and “newspolygraph” (reference: 03RU2U151C) funded by the German Federal Ministry for Research, Technology and Aeronautics (BMFTR).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 Technical Report. *arXiv preprint*. ArXiv:2412.08905 [cs].

Mubashara Akhtar, Michael Schlichtkrull, and Andreas Vlachos. 2024. Ev2r: Evaluating evidence retrieval in automated fact-checking. *arXiv preprint arXiv:2411.05375*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

DeepSeek-AI, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, and 68 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Yukang Gan, Yixiao Ge, Chang Zhou, Shupeng Su, Zhouchuan Xu, Xuyuan Xu, Quanchao Hui, Xiang Chen, Yexin Wang, and Ying Shan. 2023. Binary embedding-based retrieval at tencent. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’23, page 4056–4067, New York, NY, USA. Association for Computing Machinery.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Lucas Graves. 2018. Understanding the promise and limits of automated fact-checking. https://reut

ersinstitute.politics.ox.ac.uk/our-research/understanding-promise-and-limits-automated-fact-checking. RISJ Factsheet, February 2018.

Vishwani Gupta, Astrid Viciano, Holger Wormer, and Najmehsadat Mousavinezhad. 2023. Exploring unsupervised semantic similarity methods for claim verification in health care news articles. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 440–447, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023a. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Yinfei Liu, Weijie Liu, Junlong Li, and 1 others. 2023. Bge: Baai general embedding models. https://github.com/FlagOpen/FlagEmbedding. Accessed: 2025-05-06.

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yiling Chen, Qi Dai, Xiyang Dai, and 56 others. 2025. Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs. *arXiv preprint*. ArXiv:2503.01743 [cs].

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2023. Gpt-4 technical report. https://cdn.openai.com/papers/gpt-4.pdf. GPT-4 Turbo is an optimized variant of GPT-4, offering enhanced performance and efficiency.

OpenAI. 2024. New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates/. Accessed: 2025-05-06.

Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal question generation for fact-checking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2532–2544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sihyeong Park, Sungryeol Jeon, Chaelyn Lee, Seokhun Jeon, Byung-Soo Kim, and Jemin Lee. 2025. A Survey on Inference Engines for Large Language Models: Perspectives on Optimization and Efficiency. *arXiv preprint*. ArXiv:2505.01658 [cs].

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos, editors. 2024. *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*. Association for Computational Linguistics, Miami, Florida, USA.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. In *Thirty-thh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, and 3 others. 2023. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder,

and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Zhuohan Xie, Rui Xing, Yuxia Wang, Jiahui Geng, Hasan Iqbal, Dhruv Sahnan, Iryna Gurevych, and Preslav Nakov. 2025. FIRE: Fact-checking with iterative retrieval and verification. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2901–2914, Albuquerque, New Mexico. Association for Computational Linguistics.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. 2024. HerO at AVeriTeC: The Herd of Open Large Language Models for Verifying Real-World Claims. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 130–136, Miami, Florida, USA. Association for Computational Linguistics.

Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. 2022. Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words. *arXiv preprint*. ArXiv:2205.05092 [cs].

A Appendix

A.1 Prompts Collection

We used the following prompt for question generation:

> f"You are a professional fact checker. You recieve a claim from the user. Please provide a question you would ask to find out if a given claim is true, or not. Generate only one single question! The claim you need to check: {claim} \n Your Question:\n"

The prompt for predicting the SUP and REF labels was:

> f"You are a professional fact checker. You get a claim and provided evidence. Assess if the claim is supported or refuted by the evidence! Return only the result, either 'Supported' or 'Refuted'. The claim: {claim} \n The evidence: {retrieved_evidences} \n Your verdict: "