# Fathom: A Fast and Modular RAG Pipeline for Fact-Checking

**Farrukh Bin Rashid**
farrukh.rashid@unb.ca

**Saqib Hakak**
saqib.hakak@unb.ca

## Abstract

We present Fathom, a Retrieval-Augmented Generation (RAG) pipeline for automated fact-checking, built entirely using lightweight open-source language models. The system begins with HyDE-style question generation to expand the context around each claim, followed by a dual-stage retrieval process using BM25 and semantic similarity to gather relevant evidence. Finally, a lightweight LLM performs veracity prediction, producing both a verdict and supporting rationale. Despite relying on smaller models, our system achieved a new AVeriTeC score of 0.2043 on the test set, a 0.99% absolute improvement over the baseline and 0.378 on the dev set, marking a 27.7% absolute improvement.

## 1 Introduction

Misinformation and disinformation continue to pose serious challenges in today's social media landscape. Research from 2018 shows that false claims spread up to six times faster than truthful ones on platforms like Twitter (Vosoughi et al., 2018). Although manual fact-checking has played a critical role in addressing false claims, it struggles to match the speed and volume at which mis/disinformation proliferates online. As a result, automated fact-checking (AFC) has been proposed as a valuable tool to support journalists and professional fact-checkers (Vlachos and Riedel, 2014). AFC refers to the task of predicting the veracity of a claim by leveraging relevant pieces of evidence (Guo et al., 2022).

The field of automated fact-checking has progressed significantly in recent years, supported by the development of benchmark datasets such as FEVER (Thorne et al., 2018) and, more recently, AVeriTeC (Schlichtkrull et al., 2023). While FEVER introduced large-scale fact verification using synthetic claims derived from Wikipedia, it and similar datasets often suffered from limitations like shallow context, limited evidence, and temporal leakage, reducing their effectiveness on real-world claims. AVeriTeC addresses these challenges by using real claims from professional fact-checkers, annotated with fine-grained, crowdsourced evidence and question-answer pairs drawn from noisy, open-web sources.

In this paper, we present Fathom, a Retrieval-Augmented Generation (RAG) pipeline for automated fact-checking that leverages open-source, lightweight language models, using the AVeriTeC dataset. The system processes each claim in 22 seconds. First, it enriches the claim using a HyDE-style approach by generating hypothetical question-answer pairs. These are then used to retrieve candidate evidence via BM25. To refine the retrieved set, we apply semantic reranking using a dense embedding model. Finally, a lightweight LLM reasons over the selected evidence and QA pairs to classify the claim as Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking. The full implementation of our system is publicly available at `github.com/farrukhrashid1997/Fathom`.

## 2 Related Works

Schlichtkrull et al. (2024) features and discusses top-performing systems that combined large language models (LLMs) with hybrid retrieval pipelines for automated fact-checking on the AVeriTeC dataset. Rothermel et al. (2024) was the most notable with a GPT-4o-based pipeline and semantic search. Yoon et al. (2024) integrated BM25 with dense retrieval, using LLaMA-3 models for both question generation and veracity prediction. Ullrich et al. (2024) and Park et al. (2024) both relied on GPT-4, with the latter achieving the highest performance by enhancing retrieval through PDF and video text extraction.

A common pattern among top systems using



*Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER)*, pages 258–265
July 31, 2025 

the AVeriTeC dataset is the initial generation of questions or decomposition of claims into self-contained queries to guide retrieval. This step was typically powered by large LLMs such as GPT-4o or LLaMA-3.1-70B. However, several competitive systems, such as HerO (Yoon et al., 2024) with LLaMA-3-8B and Data-Wizards (Singhal et al., 2024) with Phi-3-medium, demonstrated that smaller models can also be effective for query generation.

Similarly, in the veracity prediction stage, most top-performing systems relied on powerful LLMs like GPT-4o, LLaMA-3.1-70B, and Mixtral-8x7B to reason over retrieved evidence and generate final verdicts. Notably, some systems like HerO (Yoon et al., 2024) and SynApSe (Churina et al., 2024) observed further gains by fine-tuning these LLMs for the task, suggesting that adapting large language models to the specific reasoning requirements of the AVeriTeC task can lead to measurable performance improvements.

More broadly, recent work has explored variations of RAG to improve fact verification through better evidence grounding. Khaliq et al. (2024) combines multimodal retrieval with structured reasoning via Chain of RAG and Tree of RAG, enabling step-by-step reasoning by retrieving and integrating evidence across multiple sub-questions.

Another notable use RAG is by CrAM (Deng et al., 2025), it improves fact-checking by teaching the model to focus more on reliable evidence. Instead of treating all retrieved documents equally, it adjusts how much attention the LLM gives to each one based on how credible it is.

Inspired by these developments, our system adopts a RAG-based architecture with lightweight open-source LLMs, combining efficient retrieval and reasoning to tackle the challenges of automated fact-checking.

## 3 Data

We evaluate our system on the AVeriTeC dataset (Schlichtkrull et al., 2023), a resource for automated fact-checking containing 4,568 real-world claims collected from 50 professional fact-checking organizations. Each claim is annotated with a veracity label *Supported*, *Refuted*, *Not Enough Evidence*, or *Conflicting Evidence/Cherry-picking* along with question-answer (QA) pairs grounded in web-based evidence, and a justification explaining how the evidence supports the verdict. The distribution of labels across the training and development splits is summarized in Table 1.

The AVeriTeC dataset uses a multi-step annotation process to make each claim easier to verify. First, the claims are cleaned and simplified for clarity. Then, annotators generate questions that reflect the core factual components of the claim. For each question, they retrieve supporting or refuting information from the web and record multiple answers along with the source URLs.

In the 2024 AVeriTeC shared task (Schlichtkrull et al., 2024) introduced a knowledge store, a curated collection of pre-retrieved web documents, to assist people using the dataset in evidence retrieval, eliminating the need for independent web scraping.

| **Class** | **Train** | **Dev** |
|---|---|---|
| Supported | 847 | 122 |
| Refuted | 1743 | 305 |
| Conflicting evidence/Cherrypicking | 196 | 38 |
| Not enough evidence | 282 | 35 |
| **Total** | **3068** | **500** |

Table 1: Class-wise distribution of train and dev sets of the AVeriTeC dataset.

## 4 Methodology

In this section, we present our system and describe its four key components within the RAG pipeline.

### 4.1 Claim Decomposition via HyDE-QA Generation

Several fact-checking pipelines (Park et al., 2024; Rothermel et al., 2024; Ullrich et al., 2024) have shown that generating explicit questions from claims and subsequently answering them using retrieved evidence significantly improves fact verification performance. These question-answer pairs not only guide the retrieval process but also organize the information in a way that supports LLM reasoning during classification. In parallel, the HyDE approach (Gao et al., 2023; Wang et al., 2023) has gained traction in RAG pipelines by generating hypothetical answers to queries using an LLM. These synthetic answers, when used as enriched search queries, improve semantic retrieval performance by injecting latent contextual cues. Inspired by prior work, we generate multiple plausible HyDE-style QA pairs for each claim to guide evidence retrieval.

In order to generate the question answer pairs, the model is prompted with the claim along with

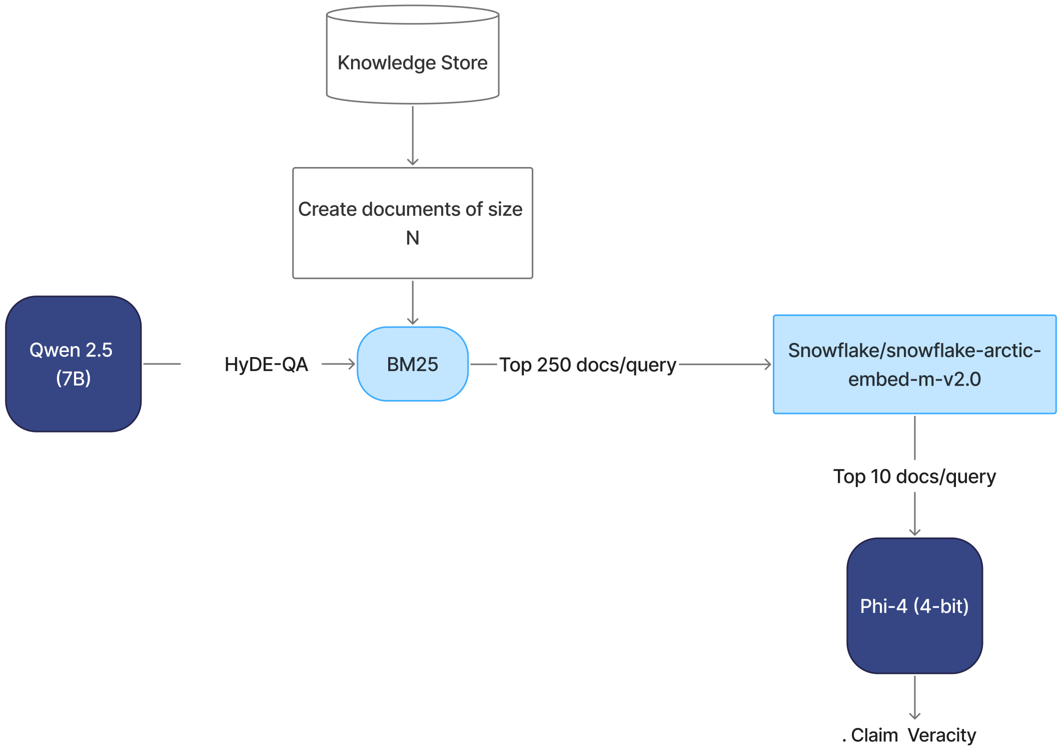


Figure 1: Overview of the Fathom system architecture.

optional metadata (speaker, date, location) and instructed to output multiple detailed QA pairs. To steer generation quality, we include handpicked few-shot examples that emphasize explicit and well-reasoned question-answer pairs. Six of the eight examples represent claims labeled as Conflicting Evidence/Cherry-picking. We chose this label primarily because such claims often involve both supporting and refuting information, which leads to more diverse and informative questions, covering multiple angles of the claim. These richer question answer pairs, in turn, guide better retrieval and support the LLM's reasoning in downstream prediction. The remaining few-shot examples include one Supported and one Refuted case. An example generation output is shown in Figure 2.

### 4.2 Evidence Retrieval

In order to retrieve relevant evidence for the claim, we adopt a hybrid retrieval strategy, combining keyword-based sparse retrieval with semantic dense re-ranking.

#### 4.2.1 Sparse Retrieval (BM25)

As in most RAG systems (Lewis et al., 2020; Ullrich et al., 2024), we perform document chunk retrieval by splitting each document into smaller passages. Consecutive sentences are grouped until the combined character length reaches a threshold of $N$. Each chunk is annotated with metadata referencing its preceding and following chunks to maintain contextual continuity when later presented to the LLM.

Given that each claim has its own associated

**Claim:** In a letter to Steve Jobs, Sean Connery refused to appear in an Apple commercial.

**Q1:** Did Sean Connery refuse to appear in an Apple commercial?
**A1:** No, because in 1998, Steve Jobs reached out to Sean Connery to star in a commercial for Apple's iMac G3. Connery agreed and filmed the commercial.

**Q2:** Was there a letter from Sean Connery refusing to appear in an Apple commercial?
**A2:** No, there is no record of such a letter. Instead, Connery accepted the invitation and starred in the commercial.

**Q3:** Did Steve Jobs request Sean Connery for a commercial?
**A3:** Yes, in 1998, Steve Jobs, the CEO of Apple, personally wrote to Sean Connery, asking him to star in a commercial for the iMac G3.

Figure 2: Example of generated QA pairs for a Refuted claim.

knowledge store, often comprising hundreds or thousands of web pages, the search space is prohibitively large for direct dense retrieval. To address this, we first apply BM25 (Harman, 1995) to narrow the candidate pool: for each generated question-answer pair, we form a query and retrieve the top $k = 250$ chunks. Results are then deduplicated across all QA pairs linked to the claim. This high-recall filtering step enables efficient downstream semantic re-ranking.

This design also ensures that the system operates within the time constraints of fact-checking a claim under one minute. BM25 acts as a lightweight and effective first-pass filter, helping balance retrieval quality with system speed.

#### 4.2.2 Semantic Re-ranking

Following the sparse BM25 stage, we employ semantic re-ranking to identify the most relevant passages from the candidate pool. To select an embedding model, we explore the MTEB benchmark (Muennighoff et al., 2022), prioritizing models under 500M parameters to meet runtime and memory constraints. We choose `Snowflake/snowflake-arctic-embed-m-v2.0` (Merrick et al., 2024), a compact model that ranks among the top performers on retrieval tasks in the MTEB leaderboard, making it well-suited for our space and time constrained setting.

Each candidate chunk is embedded into a dense vector representation that captures its semantic content. Similarly, each query is formed by concatenating the question and answer from the QA pair and encoded using the same model. We compute cosine similarity between each query and all candidate chunk embeddings, and select the top 10 highest-scoring chunks per query. The output of this step is a ranked list of 10 evidence passages for each QA pair, optimized for semantic relevance.

### 4.3 Veracity Prediction

In the final stage, the system assigns one of four veracity labels (Supported, Refuted, Not Enough Evidence, or Conflicting/Cherry-picking) based on the structured evidence retrieved earlier, along with a clear rationale for its decision.
We prompt the LLM using a structured prompt that includes:

- the original claim
- a set of generated questions (from the QA generation step)
- For each question, up to N = 8 top-ranked evidence passages (ranked by the similarity score)

Through our preliminary testing, we found that N = 8 strikes a balance between sufficient context and avoiding prompt length limitations or information overload. To encourage structured reasoning, we adopt a Chain-of-Thought (CoT) prompting strategy as shown in Figure 3. The prompt explicitly instructs the LLM to analyze the claim by reasoning step-by-step through multiple question–answer evidence pairs. Essentially, instead of making an immediate judgment, the model is guided to sequentially evaluate evidence associated with each question, reflect on its implications for the claim, and then generate an overall conclusion. This encourages the model to simulate a fact-checker's reasoning process, improving interpretability and alignment with complex veracity categories. We draw inspiration from prior work demonstrating that CoT enhances the reasoning capabilities of large language models across various tasks. (Wei et al., 2022)

After prompting the LLM, we get the final output which consists of:

- A detailed, natural-language reasoning which is grounded in the evidence
- A single veracity label, justified by the reasoning above

An excerpt of the final-stage output is shown in Figure 4, where the LLM can be seen reasoning through the provided questions and their corresponding evidence passages.

## 5 Experiments and Results

### 5.1 Experimental Details

All experiments were conducted using the **NVIDIA Quadro RTX 8000** GPU, which provided sufficient capacity for both dense retrieval and large language model inference. To ensure generalization and prevent overfitting to the dev set, final evaluation was performed on the AVeriTeC test set, an unseen split with hidden labels. For this stage, our system was deployed on an **NVIDIA A10G** GPU, as provided by the AVeriTeC shared task organizers, under standardized evaluation constraints.

Throughout the pipeline, we make design choices aimed at maximizing time efficiency without compromising output quality. In the first

| | Q (Ev2R) | Q + A (Ev2R) | New (Ev2R) | Time/claim (s) |
|---|---|---|---|---|
| Fathom | 0.2488 | **0.5137** | **0.3780** | **20** |
| Baseline | **0.3392** | 0.4404 | 0.2960 | 50 |

Table 2: AVeriTeC score on the **dev set**

| **Class** | **F1 Score** |
|---|---|
| Supported (S) | 0.6877 |
| Refuted (R) | 0.8436 |
| Not Enough Evidence (NEI) | 0.1455 |
| Conflicting/Cherry-Picking (CP/CE) | 0.0000 |
| **Accuracy** | 0.7200 |
| **Macro Avg F1** | 0.4192 |

Table 3: Per-class F1 scores, overall accuracy, and macro-averaged F1 score on the development set.

**Claim:** You are a fact-checking helpful assistant.

**Task:** Your task is to evaluate the truthfulness of a claim using associated question–answer (QA) evidence pairs, where each question has several pieces of evidence (answers). You must analyze the claim and reason step-by-step through the evidence provided. Use a chain-of-thought reasoning approach to determine the final label.

**The given claim falls into one of the following four categories:**
1. Supported
2. Refuted
3. Not Enough Evidence
4. Conflicting Evidence/Cherry-picking

**Input Format:**
Claim: <claim>
QA: <Question answer pairs>

**Output:**
Reasoning: [Use chain-of-thought reasoning on the claim based on the evidence. Incorporate evaluation of the content and optionally consider the trustworthiness or context of the source URLs.]
Label: <Supported, Refuted, Not Enough Evidence, Conflicting Evidence/Cherry-picking>

Figure 3: Prompt for the final veracity prediction step.

stage, the HyDE-QA step, we employ the open-source **`Qwen2.5-7B-Instruct`** model, running at half precision (`dtype="float16"`) to reduce memory usage and accelerate inference. We use `temperature=1.2` and `top_p=0.3` to balance output diversity and relevance where the high temperature encourages variability while the low top-p keeps sampling focused on plausible tokens to maintain coherence. To reduce latency and GPU memory load, claims are processed with a batch size of 2. These settings offer a practical trade-off between response quality and runtime performance.

In the second stage, we apply parallelism to the BM25-based sparse retrieval step. Claims are distributed across multiple CPU processes, and queries within each claim are further parallelized using multi-threading. This dual-level parallelism allows the system to retrieve evidence for many claims simultaneously, significantly reducing total time.

For semantic re-ranking, we use the **`Snowflake/snowflake-arctic-embed-m-v2.0`** embedding model. All candidate evidence chunks are encoded in batches of 128 for efficiency.

In the final veracity prediction step, we process claims in **batches of 4** using a 4-bit quantized version of Microsoft's `Phi-4` model (`jakiAJK/microsoft-phi4_GPTQ-int4`) to ensure faster inference and reduced memory usage. We select Phi-4 for its advanced reasoning capabilities, as it outperforms much larger models including LLaMA-3.3 70B and GPT-4o-mini on several reasoning-focused benchmarks such as MATH, GPQA, and HumanEval (Abdin et al., 2024). For generation, we use `temperature=0.9`, `top_p=0.7`, and `top_k=1`, a conservative setup that limits randomness while maintaining output quality. This configuration promotes stable, deterministic responses suitable for factual judgment.

### 5.2 Evaluation metrics

The evaluations metrics ensure that a correct veracity prediction is recorded only when the right evidence is retrieved to support the label.

In 2024, the AVeriTeC score was measured using the Hungarian METEOR approach (Kuhn, 1955; Banerjee and Lavie, 2005), wherein each generated QA pair is optimally matched to a reference using the Hungarian algorithm, and scored using METEOR similarity.

In 2025, the *AVeriTeC score* transitioned from Hungarian METEOR to the Ev2R atomic reference

**Claim:** In a letter to Steve Jobs, Sean Connery refused to appear in an Apple commercial.

**Reasoning:** To evaluate the claim that "In a letter to Steve Jobs, Sean Connery refused to appear in an Apple commercial," we need to examine the evidence provided in the QA pairs.

**Q1:** Did Sean Connery refuse to appear in an Apple commercial?
- **Answer 1** references multiple sources discussing a supposed rejection letter from Sean Connery to Steve Jobs, suggesting that such a letter exists. However, these sources are primarily linked to discussions about a viral letter rather than confirming its authenticity.
- **Answer 2** and **Answer 3** provide context about Apple's attempt to create a celebrity Christmas ad featuring Sean Connery, but they emphasize that the ad was never completed due to its poor quality. They do not confirm the existence of a refusal letter.
- **Answer 4** describes a letter that was widely believed to be authentic but later identified as a hoax. This suggests that while the letter was circulated, it was not genuine.

*(Q2–Q5 omitted for brevity. They further support the same conclusion.)*

**Conclusion:** The evidence consistently indicates that the letter from Sean Connery to Steve Jobs was a hoax and not a genuine refusal. While Steve Jobs did request Connery for a commercial, Connery did not actually respond with a refusal. Therefore, the claim is refuted.
**Label:** Refuted

Figure 4: Example of LLM-generated chain-of-thought reasoning with partial QA evidence (Q1–Q2 shown).

scorer (Akhtar et al., 2024). This approach uses an LLM to decompose both the predicted and reference question–answer (Q+A) pairs into sets of atomic facts. The metric then computes **Q + A (Ev2R Recall)** by measuring how many reference atomic facts are matched by those in the retrieved evidence.

The predicted veracity label is only evaluated if the **Q+A (Ev2R) Recall exceeds 0.50**, which then contributes to the final AVeriTeC score.

### 5.3 Results

We conduct all development and initial evaluation on the AVeriTeC development set. As shown in Table 2, our system achieves a higher **New (Ev2R)** AVeriTeC score of **0.3780**, outperforming the baseline. This gain is primarily driven by stronger semantic evidence retrieval, as reflected in the higher **Q+A (Ev2R)** recall score of **0.5137**.

In terms of veracity classification, Table 3 presents the F1 scores across all four classes. Our final model using the 4-bit quantized `Phi-4` performs well on the **Refuted** (0.8436) and **Supported** (0.6877) categories, which are also the most represented in the dev set. However, it struggles on the low-resource labels: **Not Enough Evidence (0.1455)** and especially **Conflicting/Cherry-picking**, where it fails to correctly classify any instance. This highlights a limitation in the veracity prediction stage, particularly for underrepresented classes that require more nuanced reasoning.

We hypothesize that one reason for this shortfall is the use of a zero-shot prompting strategy at the veracity prediction step. Incorporating a few-shot approach, especially with examples from NEI and CP/CE may help the model generalize better. Despite these challenges, the system achieves a strong overall **accuracy of 72%**, showing its reliability on majority classes while pointing to clear directions for future improvement.

To evaluate the generalization of our system, we submit predictions on the unseen AVeriTeC test set, where ground-truth labels are hidden. As shown in Table 4, our system slightly outperforming the baseline while maintaining significantly lower average runtime per claim. However, in contrast to the substantial improvement observed on the dev set, the margin over the baseline on the test set is noticeably smaller.

We hypothesize that this difference is because of our fixed chunking strategy (in step 2 of the pipeline), which segments documents using a constant token size without adopting to the semantic boundaries of the document. While this approach proved effective on the dev set, it may fail to generalize across more diverse or structurally varied claims in the test set. This is another direction for future work, to explore semantic aware chunking methods, potentially improving retrieval precision and final veracity scores.

## 6 Conclusion and Future Work

In this paper, we presented **Fathom**, our lightweight and time-efficient pipeline for evidence-based automated fact-checking, built entirely using open-source LLMs. Despite strong accuracy on majority classes such as *Refuted* and *Supported*, our results indicate notable performance gaps on underrepresented labels especially *Conflicting Evidence/Cherrypicking* and *Not Enough Evidence*. These shortcomings highlight ongoing challenges in reasoning over evidence, especially when the

|  | Q (Ev2R) | Q + A (Ev2R) | New (Ev2R) | Time/claim (s) |
|---|---|---|---|---|
| Fathom | 0.1848 | **0.3368** | **0.2043** | **22.73** |
| Baseline | **0.2723** | 0.3362 | 0.2023 | 33.88 |

Table 4: AVeriTeC Scores on the **test set**

claim is unclear or has both supporting and opposing information.

Looking forward, we identify several key directions for improving performance. First, enhancing retrieval with *semantic-aware chunking* could help the system adapt more flexibly to diverse document structures, especially on unseen data. Second, integrating *few-shot prompting* for veracity prediction may improve the model's reasoning capabilities. Finally, fine-tuning small LLMs on curated QA-veracity datasets could enable better discrimination between nuanced veracity types. Together, these enhancements may help build more robust, efficient, and interpretable fact-checking systems.

## Acknowledgment

This research is supported by NSERC RGPIN-2025-04608 and DGECR-2025-00153.

## References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

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

Satanjeev Banerjee and Alon Lavie. 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.

Svetlana Churina, Anab Barik, and Saisamarth Phaye. 2024. Improving evidence retrieval on claim verification pipeline through question enrichment. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 64–70.

Boyi Deng, Wenjie Wang, Fengbin Zhu, Qifan Wang, and Fuli Feng. 2025. Cram: Credibility-aware attention modification in llms for combating misinformation in rag. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23760–23768.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Donna K Harman. 1995. *Overview of the third text retrieval conference (TREC-3)*. 500. DIANE Publishing.

M Abdul Khaliq, Paul Chang, Mingyang Ma, Bernhard Pflugfelder, and Filip Miletić. 2024. Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models. *arXiv preprint arXiv:2404.12065*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed 2.0: Multilingual retrieval without compromise. *Preprint*, arXiv:2412.04506.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Heesoo Park, Dongjun Lee, Jaehyuk Kim, ChoongWon Park, and Changhwa Park. 2024. Dunamu-ml's submissions on averitec shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 71–76.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. Infact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana

Cocarascu, Arpit Mittal, and 1 others. 2024. The automated verification of textual claims (averitec) shared task. *arXiv preprint arXiv:2410.23850*.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36:65128–65167.

Ronit Singhal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using rag and few-shot in-context learning with llms. *arXiv preprint arXiv:2408.12060*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Herbert Ullrich, Tomáš Mlynář, and Jan Drchal. 2024. Aic ctu system at averitec: Re-framing automated fact-checking as a simple rag task. *arXiv preprint arXiv:2410.11446*.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *science*, 359(6380):1146–1151.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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. *arXiv preprint arXiv:2410.12377*.