HerO at AVeriTeC: The Herd of Open Large Language Models for Verifying Real-World Claims

Yejun Yoon[♡] **Jaeyoon Jung**[♣◇] **Seunghyun Yoon**[♠] **Kunwoo Park**[♣♡]
[♡]Department of Intelligent Semiconductors, Soongsil University
[♣]School of AI Convergence, Soongsil University
[◇]MAUM AI Inc.
[♠]Adobe Research, USA
`{yejun0382, jaeyoonskr}@soongsil.ac.kr`, `syoon@adobe.com`, `kunwoo.park@ssu.ac.kr`

Abstract

To tackle the AVeriTeC shared task hosted by the FEVER-24, we introduce a system that only employs publicly available large language models (LLMs) for each step of automated fact-checking, dubbed the **Her**d of **O**pen LLMs for verifying real-world claims (**HerO**). For evidence retrieval, a language model is used to enhance a query by generating hypothetical fact-checking documents. We prompt pretrained and fine-tuned LLMs for question generation and veracity prediction by crafting prompts with retrieved in-context samples. HerO achieved 2nd place on the leaderboard with the AVeriTeC score of 0.57, suggesting the potential of open LLMs for verifying real-world claims. For future research, we make our code publicly available at `https://github.com/ssu-humane/HerO`.

1 Introduction

Automated fact-checking is a task that predicts a claim's veracity by referring to pieces of evidence (Guo et al., 2022). Claim verification requires the retrieval of relevant information from a reliable document collection and the decision on whether the claim is supported by the known relevant information. Early research attempted to automate the fact-checking process by generating synthetic claims based on Wikipedia documents (Thorne et al., 2018; Aly et al., 2021) or collecting manually verified claims by human experts (Wang, 2017; Augenstein et al., 2019). However, most datasets suffer from critical issues such as context dependence, evidence insufficiency, and temporal leaks; these limitations made the resulting systems less applicable to the verification of real-world claims. In light of this, a recent study proposed a dataset called AVeriTeC (Schlichtkrull et al., 2023). They addressed the limitations by conducting fine-grained crowdsourced annotations for the fact-checking process.

This paper describes our system for the AVeriTeC shared task hosted by the FEVER-24 workshop (Schlichtkrull et al., 2024). Motivated by the recent advancements in large language models, we introduce a fact-checking system that utilizes LLMs for each step of evidence-based fact verification: evidence retrieval, question generation, and veracity prediction. Our system, the **Her**d of **O**pen LLMs for verifying real-world claims (**HerO**), employs publicly available LLMs without using proprietary LLMs, to ensure the transparency of the system. HerO achieved 2nd place in the shared task with an AVeriTeC score of 0.57. Given that the winning system used gpt-4o (Schlichtkrull et al., 2024), HerO's competitive performance imply the potential of open LLMs for verifying real-world claims.

2 Related Work

LLMs have achieved remarkable success in natural language understanding and generation (Brown et al., 2020; Thoppilan et al., 2022; Achiam et al., 2023). While major tech companies primarily drove the initial success, they only provided limited access to the model through an API. On the other hand, some research groups have attempted to develop open LLMs to facilitate open research. While the performance of the initial models was unsatisfactory (Zhang et al., 2022; Le Scao et al., 2023), recent models are on par with closed models and even outperform them in certain categories (Jiang et al., 2023; Dubey et al., 2024).

3 Task Definition

The AVeriTeC shared task aims to develop a fact-checking system that verifies real-world claims by retrieving evidence from the web. To verify a given claim, the system first needs to retrieve relevant information from the web documents (evidence retrieval). For each of the collected evidence, the

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 130–136
November 15, 2024 ©2024 Association for Computational Linguistics

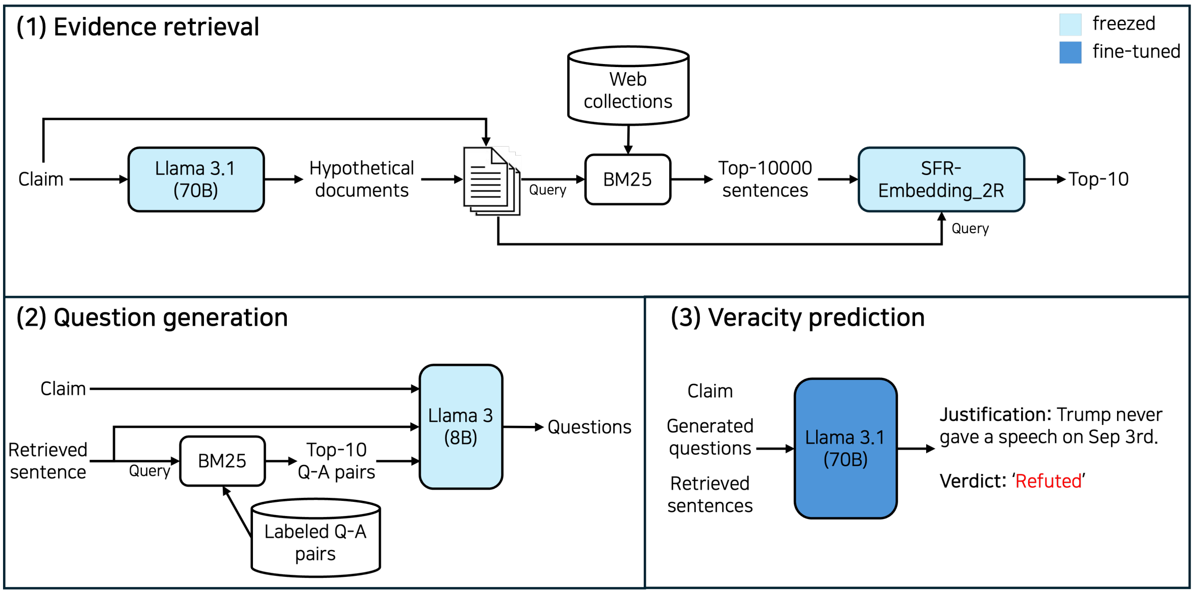

Figure 1: Inference pipeline of our system

System	Evidence Retrieval Query	Evidence Retrieval Model	Question Generation	Reranking	Veracity Prediction
Baseline	Claim	BM25	Bloom-7b	BERT-base	BERT-base
HerO	HyDE-FC (Llama-3.1-70b)	BM25 w/ SFR-embedding-2	Llama-3-8b	-	Llama-3.1-70b

Table 1: Model configurations

system may generate questions that can help verify the claim (question generation) or choose not to. The last step of the fact-checking is to verify the claim by referring to the collected information (veracity prediction). The final verdict is a four-class variable: supported, refuted, not enough evidence, or conflicting evidence/cherry-picking. Each system is evaluated using three metrics, where a higher value indicates a better score. Two metrics are the Hungarian METEOR score[1] to assess the quality of questions (Q score) and question-answer pairs (Q+A score), respectively. The overall accuracy is measured by the AVeriTeC score. Details about the task, dataset, and evaluation metrics can be found in Schlichtkrull et al. (2023) and Schlichtkrull et al. (2024).

4 Our System

This section describes our fact-checking system, the **Her**d of **O**pen LLMs for verifying real-world claims (**HerO**). Inspired by the recent progress of open LLMs (Jiang et al., 2023; Dubey et al., 2024), we only employ open LLMs for our system without using proprietary LLMs, such as gpt (Brown et al., 2020) and gemini (Team et al., 2023). Table 1 presents HerO's model configurations in comparison to the baseline system (Schlichtkrull et al., 2023). The inference pipeline of our system is illustrated in Figure 1. We use web documents provided along with the dataset as the knowledge store.

4.1 Evidence Retrieval

The first step aims to retrieve relevant sentences from the knowledge store to verify a given claim. Inspired by previous research on generative retrieval methods (Gao et al., 2023; Wang et al., 2023), we utilize an instruction-following LM to generate hypothetical fact-checking documents to augment a retrieval query. For the rest of this paper, we call this approach HyDE-FC, which stands for Hypothetical Document Embedding for Fact-Checking.

Given a claim c, we generate a set of hypothetical fact-checking documents $D = \{d_1, \ldots, d_N\}$ by prompting an instruction-following language model $f(\cdot)$ using c as an in-context sample. The used prompt for HyDE-FC is shown in Figure 2.

[1]The score uses the Hungarian algorithm (Kuhn, 1955) to find optimal matching pairs and evaluates them with the METEOR score (Banerjee and Lavie, 2005).

Please write a fact-checking article passage to support, refute, indicate not enough evidence, or present conflicting evidence regarding the claim.
Claim: *Hunter Biden had no experience in Ukraine or in the energy sector when he joined the board of Burisma.*
Passage: While Hunter Biden did not have direct experience in the energy sector or Ukraine before joining the board of Burisma, he did have ...

Figure 2: An example of the instruction prompt used for HyDE-FC and its output. The bold text is the instruction, the italic text is a claim, and the blue text indicates the model output.

We repeat the sampling process until obtaining N different documents.

Using the claim and generated documents, our retrieval pipeline employs a two-step hybrid approach that incorporates spare and dense retrieval methods. The first step is to retrieve relevant documents by BM25 (Robertson and Zaragoza, 2009). We concatenate the claim c and each document in D for building the query document q. The sparse vector for q is used to retrieve the top 10,000 relevant sentences from the knowledge store. The second step is to re-rank the 10,000 sentences by the dense retrieval method to decide the top-10 evidence candidates. The query vector v_q is obtained by averaging the embedding vectors for the claim c and every document in D by the equation 1,

$$v_q = \frac{1}{N+1}[\sum_{k=1}^{N} g(d_k) + g(c)] \tag{1}$$

where g is an embedding method.

Our best model uses llama-3.1-70b (Dubey et al., 2024) for f and SFR-embedding-2 (Meng et al., 2024) for g. N is set as 8.

4.2 Question Generation

The next step is to generate verifying questions, each of which the corresponding answer could be a retrieved sentence. We employ an instruction-following LM to generate questions for each piece of evidence. The used prompt is shown in Figure 3. We improve the baseline prompt (Schlichtkrull et al., 2023), which takes each evidence and relevant question-answer pairs from the labeled set by BM25 as in-context examples, by including a corresponding claim.

Your task is to generate a question based on the given claim and evidence. The question should clarify the relationship between the evidence and the claim

Example 1:
Claim: U.S. aid dollars sent to Ukraine under Biden's supervision went toward Burisma, where Biden's son Hunter was a board member.
Evidence: Hunter Biden was appointed to the board of Burisma.
Question: Was Hunter Biden a board member of Ukrainian energy company 'Burisma'?
...
Example 10:
Claim: Hunter Biden was paid $3 million plus $183,000 a month to be a board member of a company that a lot of people said was corrupt.
Evidence: Burisma Holdings, Ukraine's largest private gas producer, has expanded its Board of Directors by bringing on Mr. R Hunter Biden as a new director.
Question: What company is Hunter Biden a member of the board?

Now, generate a question that links the following claim and evidence:

Claim: *Hunter Biden had no experience in Ukraine or in the energy sector when he joined the board of Burisma.*
Evidence: In 2014, Hunter Biden was appointed to the board of Burisma Holdings, a Ukrainian energy company. He was reportedly paid $50,000 a month to work in an industry in which he had no previous experience.
Question: What was Hunter Biden's background or experience in the energy sector before joining the board of Burisma Holdings in 2014?

Figure 3: An example of instruction prompt and its output for question generation. The bold text indicates the instruction, the italic text is a claim, the gray text is retrieved in-context samples, and the blue text indicates the model output.

4.3 Veracity Prediction

We employ an instruction-following LM for veracity prediction. Inspired by a previous study (Wei et al., 2022), we devise a prompt that incorporates an annotator's rationale into the veracity prediction. Our best model uses the fine-tuned llama-3.1-70b-it that predicts the veracity label after generating the explanation. The top 10 question-and-answer pairs from the earlier steps are given as in-context samples along with the claim to verify.

Your task is to predict the verdict of a claim based on the provided question-answer pair evidence. The possible labels are: 'Supported', 'Refuted', 'Not Enough Evidence', 'Conflicting Evidence/Cherrypicking'. Justify your answer using the provided evidence and select the correct label.

Claim: *Hunter Biden had no experience in Ukraine or in the energy sector when he joined the board of Burisma.*

Q1: What was Hunter Biden's background or experience in the energy sector before joining the board of Burisma Holdings in 2014?
A1: In 2014, Hunter Biden was appointed to the board of Burisma Holdings, a Ukrainian energy company. He was reportedly paid $50,000 a month to work in an industry in which he had no previous experience.
...
Q10: Did Hunter Biden have any relevant experience in Ukraine or the energy sector before joining the board of Burisma?
A10: What this is all about: From the start of the inquiry, Republicans have pointed out that Hunter Biden did not have any experience in corporate governance or in the energy sector before taking the job at Burisma.

Justification: No former experience stated.
Verdict: Supported

Figure 4: An example of instruction prompt and its output for veracity prediction. The bold text indicates the instruction, the italic text is a claim, the gray text is retrieved QA pairs, and the blue text is the model output.

Query	Retrieval model	A score
Claim	BM25	0.187
	BM25 w/ SFR-embedding-2	0.26
HyDE-FC (Llama-3-8b)	BM25 w/ SFR-embedding-2	0.2745
HyDE-FC (Llama-3-70b)		0.2757
HyDE-FC (Llama-3.1-8b)		0.2751
HyDE-FC (Llama-3.1-70b)		**0.2801**
HyDE-FC (GPT-4o-mini)		0.2773

Table 2: Performance of evidence retrieval methods

Context	Model	Q score
Retrieved sentences	Baseline	0.2404
	Llama-3-8b	0.4210
	Llama-3-70b	0.4175
	Llama-3.1-8b	0.4212
	Llama-3.1-70b	0.4259
	GPT-4o-mini	0.4054
Retrieved sentences w/ Claim	Llama-3-8b	**0.4938**
	Llama-3-70b	0.4789
	Llama-3.1-8b	0.4855
	Llama-3.1-70b	0.4881

Table 3: Performance of question generation methods

5 Evaluation Experiments

In this section, we present experimental results to decide the system configuration.

5.1 Experimental Setups

In the comparison experiments, we used the development set to evaluate model performance. In addition to the Q score and Q+A score, we employed the Hungarian METEOR score to evaluate the answer quality, denoted as A score. For the comparison experiments, we used the training set for training our models and the development set for the evaluation. The training and development set were used to train our system for the submission. We used the Adam optimizer with a learning rate 2e-5, batch size 128, and 2 epochs. For LoRA, we set the rank to 128 and alpha to 256.

All the language models used in the experiments are the instruction-tuned version (e.g., llama-3.1-70b-it). For brevity, we omitted 'it' in the model identifier for the rest of the paper. For HyDE-FC, we set the LM hyperparameters as follows: maximum number of tokens as 512, temperature as 0.7, and top_p as 1.0. We used the labeled QA pairs from the training set as a data store to retrieve in-context samples for question generation. We used greedy decoding with a maximum length of 512. When an LM does not produce the verdict label, we repeated the generation with the top-2 sampling.

We ran experiments using three machines. The first has two H100 GPUs (80GB per GPU) and 480GB RAM. The second has eight H100 GPUs with 2TB RAM; the third has four NVIDIA A6000 GPUs (48GB per GPU) and 256GB RAM. The experiments were conducted in a computing environment with the following configuration: Python 3.11.9, PyTorch 2.3.1, Transformers 4.43.4, Axolotl 0.4.1, vLLM 0.5.3, and SentenceTransformers 3.0.1. HerO took approximately 6.6 hours to make 500 predictions for the development set with two H100 GPUs. It took six hours for the evidence retrieval, 25 minutes for the question generation, and 12 minutes to complete the veracity prediction.

5.2 Experimental Results

Evidence Retrieval We present evidence retrieval results on the AVeriTeC development set in Table 2. We relied on the A score as the primary metric to identify a model that can retrieve sentences that are similar to the annotated evidence.

We made three observations. First, when a claim was used as a query verbatim, applying SFR-embedding-2 to the re-ranking step boosted the performance by the A score of 0.073. Second, augmenting a query by the hypothetical document generation increased the performance. The best model, HyDE-FC with llama-3.1-70b, achieved an A score of 0.2801, 0.02 greater than the claim-only approach. Third, gpt-4o-mini was close to but slightly worse than the best open model when being used for HyDE-FC. Accordingly, HerO uses the two-step approach where SFR-embedding-2 re-ranks the top 10,000 sentences obtained by BM25; llama-3.1-70b is used to generate hypothetical fact-checking documents to augment the query.

Question Generation We present evaluation results of question generation methods in Table 3. We fixed the evidence retrieval method as the best approach to assess the effects of question generation methods. The Q score was used as a primary evaluation metric for question generation.

We made three observations. First, all the llama models achieved better Q scores than the baseline and gpt-4o-mini. Second, using the claim as an additional in-context sample boosted the generation performance significantly. The llama-3-8b model with the claim achieved a Q score of 0.4938, 0.0728 greater than its counterpart. Third, among the llama models that only use retrieved sentences as in-context samples, the latest and largest model (llama-3.1-70b) achieved the best score. However, llama-3-8b achieved the best score with the claim. Accordingly, HerO uses llama-3-8b to generate questions.

Veracity Prediction We compared veracity prediction methods using the best evidence retrieval and question generation pipelines. We evaluated three LLM-based methods: in-context learning with ten examples, instruction fine-tuning by LoRA (Hu et al., 2021), and fine-tuning the whole parameters. Table 4 shows the results. When in-context learning was used without parameter updates, the llama models outperformed gpt-4o-mini. The most significant performance gap was an accuracy of 0.14 and an AVeriTeC score of 0.112. Furthermore, the performance was boosted by instruction fine-tuning approaches. The llama-3.1-70b with the full fine-tuning approach achieved the highest AVeriTeC score of 0.578, which is the veracity prediction module for HerO.

Method	Model	Accuracy	AVeriTeC score
In-context learning	Llama-3-70b	0.628	0.494
	Llama-3.1-70b	0.54	0.422
	Gpt-4o-mini	0.488	0.382
LoRA	Llama-3-70b	0.724	0.556
	Llama-3.1-70b	0.704	0.55
Fine-tuning	Llama-3-70b	0.746	0.57
	Llama-3.1-70b	**0.752**	**0.578**

Table 4: Performance of veracity prediction methods

System	Q score	Q+A score	AVeriTeC score
TUDA_MAI_0	0.45	0.34	**0.63**
HerO	**0.48**	**0.35**	0.57
CTU AIC	0.46	0.32	0.5
Baseline	0.24	0.2	0.11

Table 5: Test set results

5.3 Test Set Results

Table 5 shows how HerO performs in the test set in comparison to the baseline and other competitive models. TUDA_MAI_0 achieved the best AVeriTeC score of 0.63, followed by HerO (0.57) and CTU AIC (0.5). Their performance gap with the existing baseline was significant. HerO achieved the best Q and Q+A scores among the top 3 models, suggesting that our question-generation approach is strong. Since HerO's performance gap with the winning system was smaller for the Q+A score than for the Q score, we suspected that our retrieval system is on par with but slightly worse than theirs. The answer score employed in our experiment could help better understand what is attributed to the performance, either retrieval or question generation.

6 Conclusion

To tackle the AVeriTeC shared task hosted by the FEVER-24, we developed HerO, a fact-checking system that employs publicly available large language models for each step of automated fact-checking: evidence retrieval, question generation, and veracity prediction. Our system achieved 2nd place in the shared task, supporting the effectiveness of open LLMs for verifying real-world claims. We release our code publicly for future research.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Graduate School of Metaverse Convergence support (IITP-2024-RS-2024-00430997) and Innovative Human Resource Development for Local Intellectualization (IITP-2024-RS-2022-00156360) programs, supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). We used an equipment supported by the NIPA (National IT Promotion Agency) under the high performance computing support program. The title and system name are hommage to research on open language models (to list a few, Jiang et al. (2023), Meng et al. (2024), and Dubey et al. (2024)), which made possible the development of our fact-checking system. Yejun Yoon and Jaeyoon Jung contributed to this work equally as co-first authors. Kunwoo Park is the corresponding author.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 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)*.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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 4685–4697. Association for Computational Linguistics.

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Rui Meng, Ye Liu, Shafiq Rayhan, Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-2: Advanced text embedding with multi-stage training.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

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. 2024. The automated verification of textual claims (averitec) shared task. In *Proceedings of* *the Seventh Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.