FZI-WIM at AVeriTeC Shared Task: Real-World Fact-Checking with Question Answering

Jin Liu[1,2], Steffen Thoma[1], and Achim Rettinger[1,3]

[1]FZI Research Center for Information Technology, Karlsruhe, Germany
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany
[3]Trier University, Trier, Germany
{jin.liu, thoma, rettinger}@fzi.de

Abstract

This paper describes the FZI-WIM system at the AVeriTeC shared Task, which aims to assess evidence-based automated fact-checking systems for real-world claims with evidence retrieved from the web. The FZI-WIM system utilizes open-source models to build a reliable fact-checking pipeline via question-answering. With different experimental setups, we show that more questions lead to higher scores in the shared task. Both in question generation and question-answering stages, sampling can be a way to improve the performance of our system. We further analyze the limitations of current open-source models for real-world claim verification. Our code is publicly available[1].

1 Introduction

Disinformation is a major concern in digital times as recent advances in generative artificial intelligence, i.e., large language models (LLMs), enable humans to create fake information on a large scale. Meanwhile, LLMs have also been integrated into automated fact-checking (AFC) systems (Chen and Shu, 2024), which have drawn lots of attention. Guo et al. (2022) summarize three stages of an AFC system: claim detection, evidence retrieval, and claim verification. Various evidence-based fact-checking datasets have been proposed for testing the systems (Thorne et al., 2018; Wadden et al., 2020; Jiang et al., 2020; Aly et al., 2021). The AVeriTeC shared task aims to fact-check real-world claims. Compared to previous fact-checking datasets, the AVeriTeC dataset (Schlichtkrull et al., 2023) utilizes question-answer (QA) pairs to tackle the complex reasoning task for real-world claims. Questioning is a natural step in the fact-checking process. The following steps involve retrieving corresponding answers and making inferences based on the QA pairs to validate the claims. Fan et al. (2020) have introduced the QABRIEF dataset, which was collected via crowdsourcing. They demonstrate that generating questions and then answering questions using open-domain question-answering can increase the accuracy and efficiency of fact-checking. With the ClaimDecomp dataset, Chen et al. (2022) show that questions to the claim can help identify relevant evidence and verify the claim with their answers.

The FZI-WIM system is composed of three stages, namely, question generation, question-answering, and claim verification. All components in the system are designed with open-source models. Given the claim and its meta information, the system first generates critical questions. A retrieval augmented generation (RAG) system is utilized to answer the generated questions with context information from the provided knowledge store. The generated QA pairs are fact-checked and filtered to tackle the potential hallucination problem. The selected QA pairs are utilized to verify the claim. We summarize our findings regarding this shared task as follows:

- More sets of distinct questions lead to better performance.
- The sampling strategy can compensate for the deficits of open-source LLMs.
- Fact-checking the RAG system is critical for getting reliable grounded answers.
- Compared to open-source models, proprietary models show significantly better performance regarding context understanding and reasoning capabilities for answering questions.

2 Background

The AVeriTeC dataset (Schlichtkrull et al., 2023) is a continuation of the previous evidence-based fact-checking dataset FEVER (Thorne et al., 2018)

[1]https://github.com/jens5588/FZI-WIM-AVERITEC

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 77–85

and FEVEROUS (Aly et al., 2021). The dataset contains real-world claims from various sources. The number of claims in the train, dev, and test set are 3068, 500, and 2215 respectively. There are five types of claims in the dataset, namely position statement, numerical claim, event/property claim, quote verification, and causal claim. The corresponding evidence has been collected from internet websites. Different from the previous dataset, which uses sentences from documents as evidence, the evidence of the AVeriTeC dataset has been formulated as QA pairs. On average, each claim in the train and dev sets has 2.6 questions. The answers can be classified into four types, boolean, abstractive, extractive, and unanswerable. Based on the QA pairs, the verification labels of the claims can be classified into supported, refuted, not enough evidence, and conflicting evidence/cherry-picking. Figure 1 shows an example from the dataset.

Claim: Donald Trump has kept his promises to voters.
Claim type: Event/Property Claim
Speaker: None
Claim date: 24-8-2020

Question 1: What promises did Donald Trump make to voters?
Answer 1 (Extractive & Abstractive): During the 2016 campaign, Donald Trump made more than 280 promises, though many were contradictory or just uttered in a single campaign event. By 2020 Trump had made a number of promises, 6 of which he had not fulfilled, including ...
Question 2: Of the promises Donald Trump made, did he fulfil any of them?
Answer 2 (Boolean): Yes.
Question 3: Has President Donald Trump kept his campaign promises to voters?
Answer 3 (Abstractive): President Trump has only kept a few of his promises.

Verification: Conflicting Evidence/Cherrypicking
Justification: QA pairs state promises kept and not kept. Claim does not state he kept all promises.

Figure 1: An example from the AVeriTeC dataset, which includes the claim, meta information, questions, answers (answer types), verification label, and justification

3 System Description

Figure 2 illustrates the three-stage pipeline of the FZI-WIM system for the AVeriTeC shared task in the test phase. In the following, we will describe the key components of each stage. The technical implementation details are presented in Appendix A.1.

3.1 Question Generator

As mentioned by (Chen et al., 2022), questions can help to identify relevant evidence. As the first component of the pipeline, raising the right questions about the claim can be critical for the final verification. Similar to the AVeriTeC dataset, the ClaimDecomp dataset (Chen et al., 2022) contains in total 1200 claims in the training, validation, and test sets while, on average, each claim has 2.7 questions. We integrate both datasets and create an instruction-tuning dataset. Besides the claim and questions, we also include the relevant meta information, such as the speaker and claim date, in the instruction dataset. We show an example of the instruction dataset in Appendix A.2.

We apply Low-rank adaption (LORA) (Hu et al., 2022), one of the parameter-efficient fine-tuning methods for LLMs, to fine-tune the existing LLM, Llama-3-70B-Instruct (AI@Meta, 2024). The concept of LORA assumes that the updates to the weights have a low intrinsic rank during the adaption of LLMs for downstream tasks. The parameter updates ΔW for a pre-trained matrix W_0 can be formulated as

$$W_0 + \Delta W = W_0 + BA, \quad (1)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$ and $r \ll \min(d, k)$ (Hu et al., 2022). Given the instruction x and target output $\{y_1, y_2, ..., y_m\}$, i.e., questions, the loss function of the training can be formulated as

$$L = \sum_{i=1}^{m} -\log(p_\theta(y_i|x, y_1, ..., y_{i-1})), \quad (2)$$

where θ represents W_0, B, A and only B and A are trainable.

With the instruction-tuned model, we first generate for each claim one set of questions greedily. With the greedy generation strategy, the model selects the token with the highest probability as its next token[2]. We further sample five sets of questions for each claim with a temperature of 0.7. With an embedding model, all-mpnet-base-v2[3] (Reimers and Gurevych, 2019), we iteratively select 2 sets from 5 sampled sets, which are most distinct from the greedy set based on the cosine similarity. Finally, each claim has three sets of questions, one greedy set, and two sampled sets.

[2] https://huggingface.co/blog/how-to-generate

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

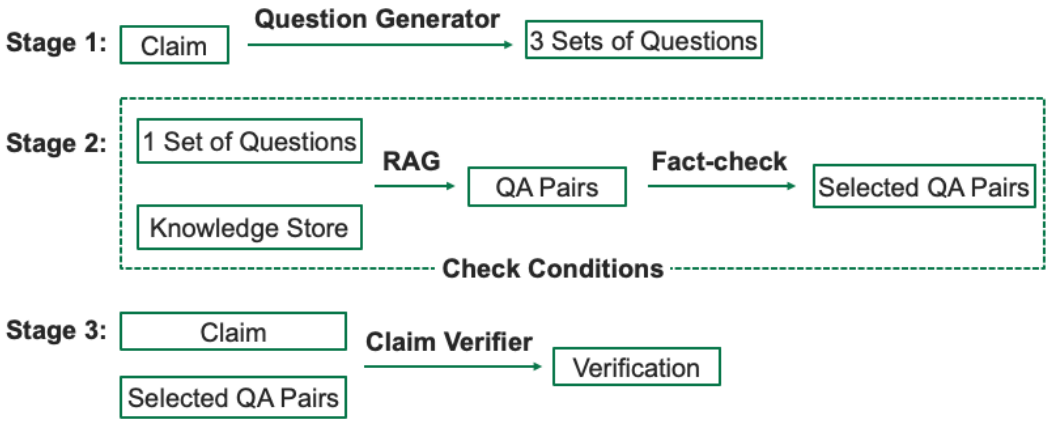

Figure 2: FZI-WIM system pipeline for the test phase in stages. In Stage 1, we first generate three sets of questions for each claim. One set of questions can contain multiple questions. Given questions and the knowledge store, our system utilizes an RAG system to generate answers to the questions. With an entailment model, the generated QA pairs are filtered. The selected QA pairs have a further conditional check. If conditions are not fulfilled, the steps in stage 2 are then repeated with another set of questions, a maximum of two repeats. Finally, an instruction-tuned claim verifier verifies the claim based on the aggregated QA pairs.

3.2 Question Answering

After generating questions for each claim, stage 2 answers these generated questions. Beginning with the greedy set of questions, the questions are answered with a retrieval augmented generation (RAG) system. We further fact-check and select answered QA pairs. We check whether the selected QA pairs fulfill the predefined conditions. If not, we then repeat the process with another sampled question set. The process is repeated at most two times.

3.2.1 RAG-based QA

Retriever After generating questions for the claims, we retrieve relevant evidence in the provided knowledge store to answer these questions. Our system only uses the provided knowledge store without querying further documents with the Google search engine. For each claim, the relevant documents are provided in the knowledge store. Various retrieval methods have been applied for documents and sentence retrieval in evidence-based fact-checking, including TF-IDF (Thorne et al., 2018), BM25 (Schlichtkrull et al., 2023), bi-encoder (Karisani and Ji, 2024), ColBERT (Khattab et al., 2021), cross-encoder (Soleimani et al., 2020), etc. Due to the limited number of relevant documents for each claim in the knowledge store, we directly apply a cross-encoder, ms-marco-MiniLM-L-12-v2[4] (Reimers and Gurevych, 2019),

[4] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

Figure 3: Input of the cross-encoder. The document is split into multiple sentence chunks so that the total length of the combination doesn't exceed 512 tokens. A sentence chunk includes about 400 to 500 tokens.

to select relevant evidence. Concretely, for each generated question, we concatenate it with the claim as the query. We then iteratively split each document into chunks so that the total length of the query and chunk pair does not exceed the maximum length of the cross-encoder, 512 tokens. Figure 3 illustrates the input of the cross-encoder. We rank the chunks based on the relevance scores predicted by the cross-encoder. For each question, we select the top 3 chunks for answering the question.

Generator With the retrieved top 3 chunks for each question, we utilize a fine-tuned LLM, Llama3-ChatQA-1.5-70B (Liu et al., 2024), to generate answers given the question and corresponding top chunks as the context. Besides the greedy generation, we sample 10 further answers with temperature 0.7 to increase the probability that the generator correctly answers the question. We show the prompt for answer generation in Appendix A.3. The candidate pool for the answer is initialized with the greedy answer. Further distinct answers from sampling are iteratively added to the candidate pool based on the similarity scores with an embedding model, all-mpnet-base-v2 (Reimers and Gurevych,

2019). In this step, one question can have multiple distinct answers. This design choice is based on our observation from experiments, that the correct answer to the question can not always be generated with the greedy decoding strategy by our generator.

3.2.2 Fact-check QA Pairs

Hallucination is a common problem of current RAG systems and it can lead to the problem that generated answers are not entailed in the source chunks. Therefore, we further add an entailment check step for generated answers. We first use few-shot learning to convert QA pairs into statements. The prompt is shown in Appendix A.4. A pre-trained natural language inference (NLI) model, bart-large-mnli[5] (Lewis et al., 2019), is used to check whether the statement is entailed in the corresponding sentence chunks. The pre-trained NLI model has three labels for (premise, hypothesis) pairs, namely refuted, not enough information (NEI), and entailed. Each statement corresponds to three sentence chunks. As soon as the statement is entailed in one sentence chunk, the corresponding QA pair will be selected. Since one question can have multiple entailed answers, i.e., statements, we select the answer with the largest entailment probability. We observe that our NLI model cannot correctly handle the entailment relationship for statements like *No information regarding ... could be found.*, which are often classified as NEI despite being entailed (supported) in the sentence chunks. So if a question has no entailed answer and there are NEI answers like *There is no information...*, *Sorry, I cannot find the answer based on the context*, etc., we also select the question with a uniform answer *No answer could be found.* for further processing. The questions that have neither entailed answers nor NEI answers are dropped.

3.2.3 Check Conditions & Aggragate

Since the fact-checking step has filtered some QA pairs, it can make the verification step difficult. We introduce two conditions to check the completeness of answers to a set of questions, namely $\frac{\#questions\ answered}{\#questions} > 0.8$ and $\frac{\#question\ answered\ with\ NEI}{\#questions\ answered} < 0.3$, where $\#questions\ answered$ represents for the number of answered questions and includes both the entailed answer and the NEI answer. If the conditions are not fulfilled, we repeat the steps in stage 2 with another set of questions.

After the maximal two times repeat, we aggregate all QA pairs for each claim. Each claim can have from one to three rounds of question answering. There can be duplicated QA pairs after aggregation. We first rank the QA pairs with a cross-encoder model based on their relevance to the claim. The QA pairs are iteratively selected with a further embedding model so that the to-be-selected pair does not exceed the similarity threshold to selected pairs. Some claims do not have any entailed or NEI answer after the third question answering round. For these claims, we use the greedy set of questions and assign *No answer could be found.* as the answer.

3.3 Claim Verification

We verify the claims with the aggregated QA pairs. Similar to the question generation process, we utilize the train and dev set to instruction-tune a pre-trained LLM, Llama-3-70B-Instruct (AI@Meta, 2024), with LORA. We show an example of the instruction dataset in Appendix A.5. We also include the justification in the target output before the verification label so that the model not only generates the verification label but also the justification. This mimics the chain-of-thought idea (Wei et al., 2022). Studies (Wang et al., 2023; Liu and Thoma, 2024) show that sampling instead of greedy decoding can improve the reasoning performance of LLMs. We sample 40 verifications for each claim and apply majority voting to derive the final verification label.

4 Evaluation

In this section, we show the performance of our proposed systems for the shared task. Besides the system in the test phase, the FZI-WIM Test, we also include the improved version in the after competition phase, FZI-WIM After Compet., for comparison. With the FZI-WIM After Compet. setup, each claim has three sets of distinct questions without conditional check described in Section 3.2.3.

4.1 Evaluation Metrics

For the shared task, both retrieved evidence and veracity predictions are evaluated. For the evidence evaluation, generated questions and answers are compared to the reference (gold questions and answers). The pairwise scoring function is defined as $f : S \times S \rightarrow \mathbb{R}$, where S is the set

[5] https://huggingface.co/facebook/bart-large-mnli

System	Q	Q+A	AVeriTeC Score
FZI-WIM Test	0.32	0.21	0.20
FZI-WIM After Compet.	0.40	0.27	0.33
Baseline	0.24	0.20	0.11
Best scores	0.49	0.35	0.63

Table 1: Overview of our systems compared to the baseline system and best scores in each category. FZI-WIM Test is our proposed system in the test phase. We further improve the system in the after competition phase with the system FZI-WIM After Compet..

of sequence tokens. The scoring function adopts the METEOR (Banerjee and Lavie, 2005) metric. The Hungarian Algorithm (Kuhn, 1955) is applied to find an optimal match between generated sequences and reference sequences (Schlichtkrull et al., 2023). A boolean function X is defined as $X : \hat{Y} \times Y \rightarrow \{0, 1\}$ to denote the assignment between the generated sequences $\hat{Y}$ and the reference sequences Y. The final score u is calculated (Schlichtkrull et al., 2023) as:

$$u_f(\hat{Y}, Y) = \frac{1}{|Y|} \max \sum_{\hat{y} \in \hat{Y}} \sum_{y \in Y} f(\hat{y}, y) X(\hat{y}, y) \tag{3}$$

The evaluation of veracity prediction uses the accuracy metric. A cut-off of $f(\hat{y}, y) \geq \lambda$ has been applied to determine whether correct evidence (concatenation of questions and answers) has been retrieved. Claims with an evidence score lower than the cut-off score λ receive veracity scores of 0. The AVeriTeC score in the shared task has a λ value of 0.25 (Schlichtkrull et al., 2023).

4.2 Results

Table 1 shows the performance of our proposed systems compared to the baseline system and the best scores in each category. After the competition, we further improved our system with more questions (FZI-WIM After Compet.). Concretely, we remove the conditional check step and further repeat stage 2 twice for every claim. This means each claim has three sets of questions and three rounds of question answering. With more questions, we can observe significant performance improvement regarding three metrics. In the following, we give a detailed analysis of our system regarding question generation & answering and claim verification.

4.3 QA Analysis

Table 2 shows the statistics of three different setups for selecting QA pairs. In the Greedy setup, the selected QA pairs for each claim are aggregated only with the greedy set of questions. In the FZI-WIM Test setup, with the conditional check, 1405 claims have utilized one set of questions, 365 claims with 2 sets of questions, and 445 claims with three sets of questions to select QA pairs. In the FZI-WIM After Compet. setup all 2215 claims have three sets of questions to select QA pairs. From the results, we can observe that more sets of different questions improve the scoring of both question and QA pairs. This is partly because we have not retrieved extra documents outside the knowledge store, which can cause questions to be not properly answered. There are various ways to ask critical questions for each claim, i.e., various reasoning possibilities. More sets of different questions can increase the probability of matching the gold questions. In the following, we give a further analysis regarding each component in our question-answering pipeline, with a focus on the deficits that cause errors.

Retriever We have directly applied a cross-encoder model to select relevant chunks from the document corpus. Compared to other methods, e.g., TF-IDF, dual-encoder, etc., the advantage of the cross-encoder is the retriever performance, and the disadvantage is the computing time. Another limitation of the cross-encoder model is the input length, in our case a maximum of 512 tokens. The incomplete context information can lead to misleading answers, especially adversarial information, i.e., misinformation or satire exists in the context.

Generator We have utilized Llama3-ChatQA-1.5-70B (Liu et al., 2024) from Nvidia to generate answers with a zero-shot setup. For a question, the corresponding context combined of the top 3 sentence chunks, normally includes around 1500 tokens. Hallucination and insufficient understanding of questions and contexts are two major reasons leading to wrong answers. We observe that with the greedy generation, the model cannot always come to the correct answer. We further sample 10 answers with a temperature of 0.7 for each question. Table 3 shows the distribution of answer sources. The statistics show the necessity of sampling besides the greedy generation.

Fact-check The difference between the number of total questions and answered questions in Table 3 reflects the number of dropped questions under

Setup	#Total Questions	#Selected QA	NEI (%)	Q	Q+A
Greedy Set of Questions	5004	3846	17.57	0.28	0.18
FZI-WIM Test	8212	5574	16.02	0.32	0.20
FZI-WIM After Compet.	16696	10048	18.68	0.40	0.27

Table 2: Comparison of different setups for QA pairs selection, including the numbers of total generated questions and selected QA pairs, percentage of the NEI answer in selected QA pairs, and the resulting question scores, question + answer scores.

Setup	#Total Questions	#Answered	Greedy / Sampling (%)
Greedy Set of Questions	5004	4381	74.30 / 25.70
FZI-WIM Test	8212	7004	69.20 / 30.80
FZI-WIM After Compet.	16696	14512	68.54 / 31.46

Table 3: Distribution of answers, including entailed and NEI answers, among greedy generation and sampling under different setups.

System	Greedy	Sampling
FZI-WIM Test	0.1991	0.1959
FZI-WIM After Compet.	0.3314	0.3336

Table 4: Comparison of AVeriTeC scores under greedy generation and sampling strategies for claim verification. The same QA pairs are used for each system with two strategies.

each setup. The dropped questions have neither entailed answers nor NEI answers, which shows the necessity of fact-checking the RAG system in the pipeline. We have utilized a pre-trained discriminative NLI model, bart-large-mnli (Lewis et al., 2019), with a maximum input length of 1024 tokens. Existing pre-training datasets for NLI, i.e., MNLI, SNLI, etc., have normally short contexts. Given the trend of growing context length in the current RAG systems, reliable entailment-check at the document level can be interesting for future research.

4.4 Claim Verification

The claim is verified with an instruction-tuned model. In the submitted systems, we have sampled 40 verifications for each claim and applied majority voting to select the final label. With the same instruction-tuned model and QA pairs, we generate the verification greedily for comparison. Table 4 shows the verification performance of greedy generation and sampling. The performance difference regarding the AVeriTeC score is negligible between the two strategies. This can be partly attributed to the final AVeriTeC scoring function. We can only conclude the greedy generation and sampling for claims, whose corresponding QA pairs compared to gold QA pairs have exceeded the cut-off threshold of 0.25, make a small difference. For claims with QA scores smaller than 0.25, which are not necessarily wrong, the effect of sampling compared to the greedy generation is not reflected in the AVeriTeC scores.

4.5 Open-source VS Proprietary Models

We have observed the current bottleneck of our pipeline lies in the generator, which utilizes an open-source LLM Llama3-ChatQA-1.5-70B (Liu et al., 2024) as the backbone to answer questions. We conduct further experiments and replace the open-source LLM with a proprietary model, namely GPT4-Turbo from OpenAI [6]. Concretely we apply the same question generator, retriever, and claim verifier as shown in Figure 2. Only the generator is replaced with GPT4-Turbo. Due to the budget constraint, we evaluate the model only on the dev set and generate the answers greedily (temperature 0) without sampling. We have not fact-checked (entailment check) the answers from GPT4-Turbo, which is generally wordy compared to the open-source generator and makes the entailment check difficult. We have utilized maximal two sets of distinct questions. For comparison, we select the FZI-WIM After Compet. system, which utilizes three sets of distinct questions for each claim. The results are shown in Table 5. The Q+A scores in the table demonstrate significantly better performance of GPT4-Turbo than the open-source generator. Our manual investigation shows also that GPT4-Turbo has better context understanding and reasoning capabilities, especially in adversarial cases.

[6] https://openai.com

Setup	#Selected QA	Q	Q+A	AVeriTeC Score
FZI-WIM After Compet.	2266	0.41	0.26	0.29
GPT4-Turbo (1 Set Questions)	1096	0.32	0.22	0.24
GPT-4 Turbo (2 Sets Questions)	2372	0.42	0.30	0.45

Table 5: Comparison between open-source and proprietary LLMs as the generator for answering questions on the dev dataset. FZI-WIM After Compet. utilizes all three sets of questions.

5 Conclusion & Outlook

In this paper, we have described the FZI-WIM system for the AVeriTeC shared task, which aims to tackle the real-world claim verification problem. The complex reasoning problem in fact-checking is tackled via question-answering. For each claim, we first generate relevant critical questions. Based on the provided knowledge store, the questions are answered with an RAG system. Considering the hallucination problem in RAG systems, we fact-check the generated QA pairs to ensure the answers are entailed in the source texts. We show that more questions, i.e., more question-answering rounds, lead to better model performance. The claim verification is based on the selected QA pairs.

Generally, our current systems need a large amount of computing. The improvement of the efficiency with open-source models is needed for the real-world scenario. Compared to proprietary models, our generator in the RAG system is not robust enough against adversarial contexts, e.g., misinformation, satire, etc. Further enhancement of the robustness can be a promising research direction.

6 Acknowledgments

This work was carried out with the support of the German Federal Ministry of Education and Research (BMBF) within the project "DeFaktS" (Grant 16KIS1524K). This work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

Limitations

Due to the limited time for developing the systems in the test phase, our systems have only used the provided knowledge store without searching for extra relevant documents related to our questions. Extra search can make a big difference for certain steps, e.g., the repeated processes in stage 2. With extra search, the times of repeats can be reduced. To achieve the best performance our current systems have always selected better-performed open-source models, e.g., cross-encoder, LLMs, etc., which normally have a larger size. This leads to the fact that our systems require a large amount of computing. In the future, we will focus on the trade-off of performance and efficiency for real-world fact-checking systems.

References

AI@Meta. 2024. Llama 3 model card.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7147–7161, Online. 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. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Payam Karisani and Heng Ji. 2024. Fact checking beyond training set. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2247–2261, Mexico City, Mexico. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jin Liu and Steffen Thoma. 2024. FZI-WIM at SemEval-2024 task 2: Self-consistent CoT for complex NLI in biomedical domain. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1269–1279, Mexico City, Mexico. Association for Computational Linguistics.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. *arXiv preprint arXiv:2401.10225*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval*, pages 359–366, Cham. Springer International Publishing.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. 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*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch FSDP: experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860.

A Appendix

A.1 Implementation details

Instruction-tuning We have applied Fully Shared Data Parallel (FSDP) from Meta AI (Zhao et al., 2023) for the instruction-tuning of question generation and claim verification models. The training script is based on llama-recipes[7] with two 4×Nvidia-H100 nodes. The dev sets are included for fine-tuning to make predictions on the final test set. For question generation, we have fine-tuned for 5 epochs and claim verification for 3 epochs.

Model inference We have applied transformers library[8] for inference. For the greedy generation, we set the parameter *do_sample* as false. For sampling, we set *temperature* as 0.7 and *top_k* as 50.

A.2 Example for instruction-tuning question generator

Figure 4 shows an example of the instruction-tuning dataset for the question generator.

[7] https://github.com/meta-llama/llama-recipes

[8] https://github.com/huggingface/transformers

You are a fact-checker and your task is to generate critical questions for verifying the following claim.
Claim date: 25-8-2020
Claimer: Pam Bondi
Claim: Hunter Biden had no experience in Ukraine or in the energy sector when he joined the board of Burisma.
Questions: Did Hunter Biden have any experience in the energy sector at the time he joined the board of the Burisma energy company in 2014? Did Hunter Biden have any experience in Ukraine at the time he joined the board of the Burisma energy company in 2014?

Figure 4: An example of the instruction dataset for fine-tuning an LLM to generate questions. The prompt ends with "Questions: ". The questions are the target output for fine-tuning the LLM.

A.3 Prompt for question-answering

Figure 5 shows the prompt for question-answering.

System: This is a chat between a user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions based on the context. The assistant should also indicate when the answer cannot be found in the context.

GSK does not own Pfizer and or the Wuhan biological laboratory You have sent us an Instagram message with these and other misleading and false relation ...

Disclosure: The Open Society Foundations and Bill and Melinda Gates Foundation are among Africa Check's funders, which together provided 21% of our income in 2019 ...

Rumor – Facts list shows that the Wuhan Laboratory is owned by Glaxo, Pfizer, has connections with foreign companies and receives money from George Soros and Bill Gates ...

User: Please give a full and complete answer for the question. **Who owns GlaxoSmithkline?**

Assistant:

Figure 5: Prompt template for answering the question given the top 3 chunks, adopted from Liu et al. (2024). The top 3 chunks in the context are ordered reversely.

A.4 Few-shot prompt for converting QA pairs to statements

Figure 6 shows the few-shot examples to convert QA pairs to statements.

A.5 Example for instruction-tuning claim verifier

Figure 7 shows an example of the instruction dataset for the claim verification.

Your task is to convert question answer pairs into statements. In the following there are some example showing how to convert question answer pairs into statements.

Question: What resolutions did Biden support in favor of US intervention in Iraq?
Answer: He supported the H.J.Res.114 - Authorization for Use of Military Force Against Iraq Resolution of 2002 107th Congress (2001-2002)
Statement: Joe Biden supported the H.J.Res.114 - Authorization for Use of Military Force Against Iraq Resolution of 2002 107th Congress (2001-2002)

Question: How much of their national budget did the Kenyan judiciary receive in 2021?
Answer: Budget speeches for 2020/21 show the judiciary received 0.6% of the national budget.
Statement: Budget speeches for 2020/21 show the Kenyan judiciary received 0.6% of the national budget.

Question: Should counties be chasing the 10% spending target or should it be done at a national level?
Answer: No answer could be found.
Statement: No answer could be found regarding whether counties should be chasing the 10% spending target or if it should be done at a national level.

Question: Did Hunter Biden have any experience in the energy sector at the time he joined the board of the Burisma energy company in 2014
Answer: No
Statement: Hunter Biden didn't have any experience in the energy sector at the time he joined the board of the Burisma energy company in 2014.

Figure 6: Few-shot prompt for converting QA pairs to statements.

Your task is to verify the claims based on the context information in format of question answer pairs. Verify the claim with justification using the following labels: Supported, Refuted, Not Enough Evidence, Conflicting Evidence/Cherrypicking.

Claim: Hunter Biden had no experience in Ukraine or in the energy sector when he joined the board of Burisma.
Question 1: Did Hunter Biden have any experience in the energy sector at the time he joined the board of the Burisma energy company in 2014
Answer 1: No
Question 2: Did Hunter Biden have any experience in Ukraine at the time he joined the board of the Burisma energy company in 2014
Answer 2: No

Justification: No former experience stated.
Label: Supported

Figure 7: An example of the instruction dataset for fine-tuning an LLM to verify the claims. The prompt ends with "Answer 2: No ". The justification and label are the target output.