RAG-Fusion Based Information Retrieval for Fact-Checking

Yuki Momii, Tetsuya Takiguchi, Yasuo Ariki
Graduate School of System Informatics, Kobe University
235x075x@gsuite.kobe-u.ac.jp, {takigu, ariki}@kobe-u.ac.jp

Abstract

Fact-checking involves searching for relevant evidence and determining whether the given claim contains any misinformation. In this paper, we propose a fact verification system based on RAG-Fusion. We use GPT-4o to generate questions from the claim, which helps improve the accuracy of evidence retrieval.

Additionally, we adopt GPT-4o for the final judgment module and refine the prompts to enhance the detection accuracy, particularly when the claim contains misinformation. Experiment showed that our system achieved an AVeriTeC Score of 0.3865 on the AVeriTeC test data, significantly surpassing the baseline score of 0.11.

1 Introduction

In recent years, misinformation has become easier to spread online (Guo et al., 2022). Consequently, to prevent its spread, the demand for automated fact-checking, which automatically detects unreliable information has significantly increased (Nakov et al., 2021). Fact-checking involves searching for information necessary for verification (evidence) from reliable external databases, and determining the truthfulness of given claim based on that information (Zhou et al., 2019).

There are various fact-checking datasets, with unstructured data like text (Thorne et al., 2018; Schuster et al., 2021) and structured data like tables (Wenhu Chen and Wang, 2020; Aly et al., 2021) or knowledge graphs (Kim et al., 2023). Generally, these datasets include a claim, the evidence that needs to be searched to verify the claim, and a label indicating the judgment.

For example, in FEVER (Thorne et al., 2018), claims need to be classified into three labels: "Supported", "Refuted", or "Not Enough Information". Numerous systems have been proposed (DeHaven and Scott, 2023; Krishna et al., 2022; Liu et al., 2020), and the accuracy of this three-class classification has reached nearly 0.8[1]. However, the claims included in these datasets are created from sources like Wikipedia for specific purposes, and they differ from the claims that journalists actually verify. There is a dataset that include real-world data (Wang, 2017), but they face the issue of not providing sufficient evidence necessary for judgment (Schlichtkrull et al., 2023).

In this Shared Task, AVeriTeC(Schlichtkrull et al., 2023) has been newly created. In AVeriTeC, the evidence is based on information collected from the web and is provided in a Question-Answer pair format by human annotators. The judgment labels are: "Supported", "Refuted", "Not Enough Evidence (NEE)", and "Conflicting Evidence/Cherry-picking". Additionally, for each claim, the reasons why annotators assign the judgment labels are annotated.

The system needs to extract evidence from documents obtained through web searches or from documents provided by the organizers as web search results, and then predicts the claim label. The claim is considered correctly judged only if the necessary evidence is appropriately retrieved, and the final judgment label is correctly predicted.

In this paper, we designed the system shown in Figure 1 to improve the AVeriTeC baseline. The baseline system primarily used BM25 (Robertson and Zaragoza, 2009) for evidence collection, but this method does not allow for searching based on the meaning of the claim or web document. Therefore, we perform searches using embedding vectors with stella_en_400M_v5[2]. We generate embedding vectors for the claim and the document, and collect 50 documents related to the claim based on their similarity.

Next, inspired by RAG-Fusion (Rackauckas,

[1] https://competitions.codalab.org/competitions/18814

[2] https://huggingface.co/dunzhang/stella_en_400M_v5

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 47–54
November 15, 2024 ©2024 Association for Computational Linguistics

2024), we use GPT-4o to generate three questions from the claim that are needed to search for the evidence. For each of these generated questions, we select three answer sentences from the previously collected 50 documents. These Question-Answer pairs collected through this procedure are input into GPT-4o along with the claim for the final judgment in verdict inference.

The proposed fact-checking system achieved an AVeriTec score of 0.3865 on the test data.

2 System Description

The system we developed is structured in three phases similar to (Gi et al., 2021): Document Retrieval, Question Generation and Sentence Retrieval and Verdict Inference. **Document Retrieval**: Since the document set provided by the organizers is vast, this phase selects documents related to the claim. **Question Generation and Sentence Retrieval**: Referring to the RAG-Fusion method, questions for information retrieval are generated using GPT-4o from the claim. Subsequently, the sentences that answer these generated questions are retrieved from the sentences contained within the documents selected in the Document Retrieval phase. **Verdict Inference**: Using GPT-4o, which has high inferential capabilities, a judgment is made based on the obtained Question-Answer pairs and the claim. We use GPT-4o via OpenAI API[3].

2.1 Document Retrieval

The AVeriTec dataset provides an average of 999.3 documents per claim, and splitting them into sentences would require extensive resources. Therefore, the target of this phase is to narrow down the candidates at the document level.

In the baseline system, all documents related to a claim were split into sentences, and relevant sentences for each claim were retrieved primarily using BM25. However, this approach doesn't account for paraphrasing or semantic similarity, limiting its search performance. Therefore, we use stella_en_400M_v5 to perform searches for the related documents using embedding vectors. At the time of writing this paper, stella_en_400M_v5 was the highest-performing model under 1B on the MTEB leader-board[4]. Given the vast amount of document to be processed in this dataset, a lightweight model was chosen. Each claim and the documents provided for that claim are converted into embedding vectors, and relevant documents are selected based on similarity. (The prompt used for embedding claim was *s2p_query* (sentence to passage query). When we use stella_en_400M_v5 for embedding search sentence, we can select *s2p_query* or *s2s_query* (sentence to sentence query) depending on our purpose).

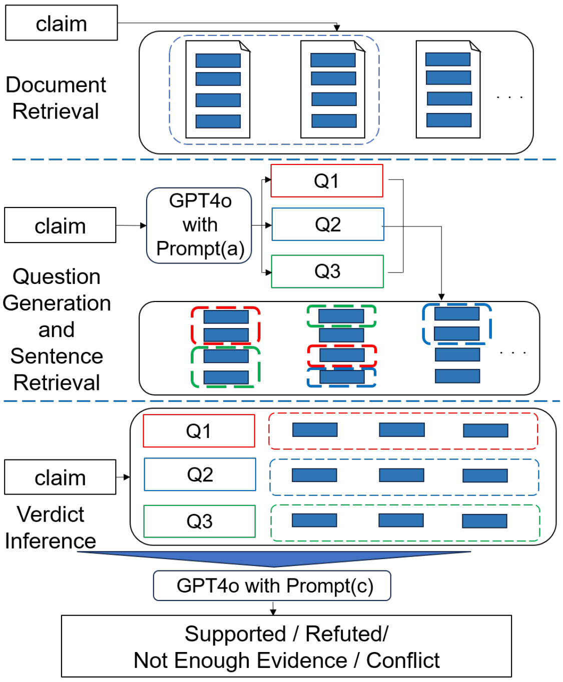


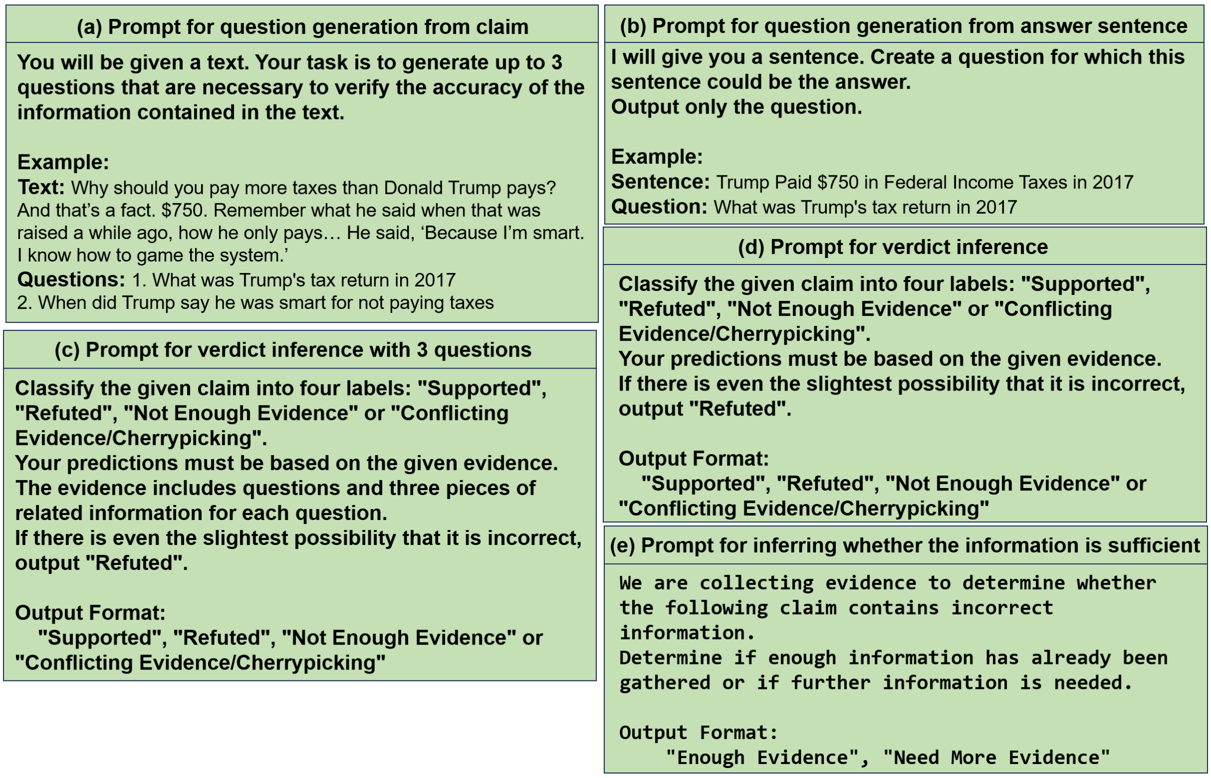
Figure 1: System Overview: Document Retrieval, Question generation and Sentence Retrieval, and Verdict inference. In Document Retrieval, 50 documents are searched. In Sentence Retrieval, up to 3 questions are generated, and for each question, 3 candidate answers are retrieved.

2.2 Question Generation and Sentence Retrieval

After narrowing down documents with Document Retrieval, the document is split into sentences to search for more critical information. The URL of each sentence remains the same as that of the original document before splitting.

The simplest approach is to convert both the claim and each sentence into embedding vectors then retrieve the most similar sentences. On the other hand, a method called RAG-Fusion (Rackauckas, 2024) has been proposed. RAG is a system that searches for relevant information in response to a user's input and uses both the input and the retrieved information to generate a response through

[3] https://openai.com/api/
[4] https://huggingface.co/spaces/mteb/leaderboard



(a) Prompt for question generation from claim

You will be given a text. Your task is to generate up to 3 questions that are necessary to verify the accuracy of the information contained in the text.

Example:
Text: Why should you pay more taxes than Donald Trump pays? And that's a fact. $750. Remember what he said when that was raised a while ago, how he only pays… He said, 'Because I'm smart. I know how to game the system.'
Questions: 1. What was Trump's tax return in 2017
2. When did Trump say he was smart for not paying taxes

(b) Prompt for question generation from answer sentence

I will give you a sentence. Create a question for which this sentence could be the answer.
Output only the question.

Example:
Sentence: Trump Paid $750 in Federal Income Taxes in 2017
Question: What was Trump's tax return in 2017

(c) Prompt for verdict inference with 3 questions

Classify the given claim into four labels: "Supported", "Refuted", "Not Enough Evidence" or "Conflicting Evidence/Cherrypicking".
Your predictions must be based on the given evidence.
The evidence includes questions and three pieces of related information for each question.
If there is even the slightest possibility that it is incorrect, output "Refuted".

Output Format:
"Supported", "Refuted", "Not Enough Evidence" or "Conflicting Evidence/Cherrypicking"

(d) Prompt for verdict inference

Classify the given claim into four labels: "Supported", "Refuted", "Not Enough Evidence" or "Conflicting Evidence/Cherrypicking".
Your predictions must be based on the given evidence.
If there is even the slightest possibility that it is incorrect, output "Refuted".

Output Format:
"Supported", "Refuted", "Not Enough Evidence" or "Conflicting Evidence/Cherrypicking"

(e) Prompt for inferring whether the information is sufficient

```
We are collecting evidence to determine whether
the following claim contains incorrect
information.
Determine if enough information has already been
gathered or if further information is needed.

Output Format:
    "Enough Evidence", "Need More Evidence"
```

Figure 2: Prompts designed for GPT-4o. In our final system, we use (a) and (c). The other prompts are used only for performance evaluation purposes.

a language model (Gao et al., 2024). The concept of retrieving relevant information and using it in subsequent processing is similar to fact-checking.

RAG-Fusion is a method proposed to enhance the retrieval performance of RAG. Instead of directly searching with the user's input, it conducts the search using multiple questions generated from user's input by LLMs (Large Language Models) and re-ranks the external information based on the search results. This approach allows for a broader perspective in the search process compared to searching directly with the user's input, potentially improving search accuracy.

In this study, we focus on RAG-Fusion's ability to retrieve diverse information through search using multiple questions. Using the prompt shown in Figure 2(a), three questions were generated from the claim using GPT-4o to search for information necessary for judgment. At this time, the claim most similar to the target claim was retrieved from the training data (using stella_en_400M_v5), and questions were copied from the evidence annotated to that claim to as the one-shot example included in the prompt. (When experimenting with validation data (500 claims), the claim is retrieved from the training data (3068 claims); when experimenting with test data (2215 claims), it's retrieved from both the training and validation data.)

For each question, three appropriate answers were retrieved, just as before, using stella_en_400M_v5. However, when stella is used to search for similar claims to generate questions, it is set to *s2s_query*; when searching for answers, it is set to *s2p_query*.

2.3 Verdict inference

In the final judgment, based on the created Evidence (Question and Answer), the system must classify the claim into one of four categories: "Supported", "Refuted", "Not Enough Evidence" or "Conflicting Evidence/Cherry-picking". We used GPT-4o for this judgement. In Fact-checking, the most critical error to avoid is mistakenly classifying a "Refuted" claim as another label. Therefore, the prompt includes the instruction: "If there is even the slightest possibility that it is incorrect, output 'Refuted'." The prompt is shown in Figure 2(c).

3 Result

In this chapter, we explain the results at each phase of the system. To consider improving search accuracy, we report the experimental results using a validation dataset (containing 500 claims) where the correct evidence has been distributed. Additionally, when using GPT-4o, the temperature is set to 0 to ensure the reproducibility of the experiments.

3.1 Document Retrieval Result

To verify how many documents could be retrieved necessary for judgment, we utilize the annotated URLs. We counted the number of claims for which the search was successful by comparing the URLs of documents annotated as the necessary sentences for judgment with the URLs of the documents retrieved through embedding vectors (up to a maximum of 500 claims in the validation data). The verification is conducted under two settings: when all the correct URLs are retrieved (All) and when at least one correct URL is retrieved (Easy).

We compared two document retrieval methods: one that uses embedding vectors of claims and documents as described in 2.1, and another that uses the questions generated by the method described in 2.2. The questions generated in 2.2 can also be used for document retrieval. Therefore, each question is converted into an embedding vector and used for document retrieval. We compared whether it is better to use the claim itself or the question generated from the claim for document retrieval.

The search results are shown in Table 1. In the table, “top k” refers to the top k results for each question in the **question-based** search. In other words, the top 25 for each question retrieves the same number of documents as the top 75 in the **claim-based** search (25×3=75). However, in the baseline system of (Schlichtkrull et al., 2023), documents were divided into sentences before the search, so a comparison at this stage cannot be made.

The comparison between the top 75 in claim-based search and the top 25 in question-based search in Table 1 shows that claim-based search yields higher accuracy. Of course, if we increase the top k, search accuracy improves naturally. However, considering computational costs, we decided to retrieve the top 50 documents in claim-based search for this time.

Method	Top k	Easy	All
Claim	Top 75	283	90
	Top 50	247	78
	Top 25	187	54
Question	Top 75	313	115
	Top 50	295	100
	Top 25	242	72

Table 1: Document Retrieval Result

Method	Top k	Easy	All
Base	Top 10	51	14
	Top 3	33	8
	Top 1	17	4
Claim	Top 10	94	27
	Top 3	50	15
	Top 1	26	8
Question	Top 10	143	36
	Top 3	79	19
	Top 1	44	13

Table 2: Sentence Search Result

Method	Q	A	Q+A
Claim (Top 3)	0.3063	0.1814	0.2258
Question (Top 1)	0.3898	0.1699	0.2436

Table 3: Evidence evaluation score of Sentence Search Result

3.2 Sentence Retrieval Results

We compare the performance of sentence retrieval using BM25 at the baseline and retrieval using embedding vectors. In the original baseline, a re-ranker was employed, but the results before introducing the re-ranker are shown for performance comparison. For retrieval using embedding vectors, we employ two methods: one based on the RAG-Fusion method explained in 2.2 and another based on the claim-based retrieval method. Similar to the comparison in 3.1, the top k retrieval results using the question correspond to the number of documents retrieved in the top 3k using the claim.

For evaluation, we report scores based on whether all correct URLs were retrieved or at least one correct URL was retrieved, using the URLs obtained from the retrieved sentences and the correct URLs. The results are shown in Table 2.

When comparing the top 1 in the question-based retrieval and the top 3 in the claim-based retrieval, the retrieval performance is nearly equivalent. Both methods yield higher scores than the baseline. Of course, this evaluation simply calculates the score based on URLs, so there might be cases where an unrelated sentence from the same document as the correct answer is retrieved. Therefore, we also report the evidence evaluation score used in this Shared Task. The evidence evaluation score is calculated as following:

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

Here, X is a boolean function denoting the assignment: $\hat{Y}$ ×Y→{0, 1}. $\hat{Y}$ is generated se-

Method	Q	Q+A	Label Accuracy	AVeriTec Score (.1, .2, .25)		
Claim (Top 3)	0.3063	0.2258	0.568	0.528	0.336	0.198
Question (Top 1)	0.3898	0.2436	0.612	0.588	0.384	0.264
Question (Top 3)	0.3898	0.2757	0.692	0.676	0.524	0.38
Gold Evidence	1.0	1.0	0.858	0.858	0.858	0.858

Table 4: Results of claim-based method and question-based method on the validation dataset. AVeriTec Scores are conditioned on correct evidence (Q+A) at λ=(0.1, 0.2, 0.25)

quences and Y is the reference sequences. f is a pairwise scoring function: $\hat{Y} \times Y \rightarrow \mathbb{R}$.

In the Shared Task, two scenarios are evaluated: one where only the question from the QA pair provided as necessary information for the judgment is used, and another where the combination of the question and the answer is used. In this paper, to compare performance in more detail, we also included the scenario where only the answer is used.

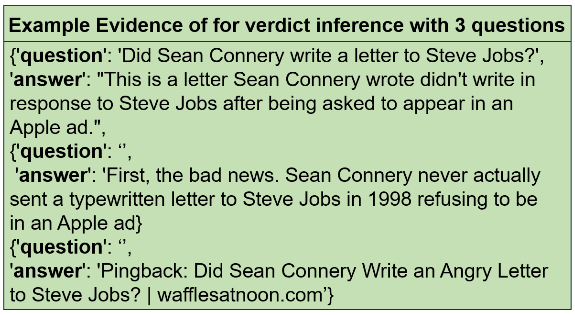
In retrieval with the claim and the baseline, the relevant sentences associated with the claim have been retrieved at this point. Consequently information corresponding to the answer has been retrieved. However, the part corresponding to the question has not yet been created. Therefore, we used GPT-4o to generate a question that would match the retrieved sentence as an answer. In this way, we created Question-Answer pairs in the same format as the correct evidence provided for the judgment. The prompt used is shown in Figure 2(b), and the scores are shown in Table 3.

The comparison between claim-based and question-based approaches shows that the pre-creation of questions yields higher Question scores, which in turn improves the Question+Answer scores. On the other hand, the score for the answer alone is slightly higher when using the approach of retrieving with the claim alone and then generating the question afterward. Since this evaluation metric only assesses sequence match, it is difficult to determine superiority at this point. Therefore, we decided to calculate the performance of both methods in the next Verdict Inference and select the approach with higher accuracy.

3.3 Verdict Inference Result

For the final evaluation, we employed GPT-4o. Using the prompt shown in Figure 2(d), we compare the results of Question Top 1 and Claim Top 3.

In the Shared Task, a judgment was considered correct only when the evidence evaluation score (Eq. (1)) exceeded a certain threshold and the final judgment was correct (AVeriTeC Score). However, the AVeriTeC Score is solely based on sequence matching and does not account for the meaning of the sentences. Moreover, it is possible to retrieve information useful for judgment outside of the correct evidence. This indicates that the evidence retrieval may not have been adequately evaluated by AVeriTeC Score.



Example Evidence of for verdict inference with 3 questions

{'**question**': 'Did Sean Connery write a letter to Steve Jobs?',
'**answer**': "This is a letter Sean Connery wrote didn't write in response to Steve Jobs after being asked to appear in an Apple ad.",
{'**question**': '',
'**answer**': 'First, the bad news. Sean Connery never actually sent a typewritten letter to Steve Jobs in 1998 refusing to be in an Apple ad}
{'**question**': '',
'**answer**': 'Pingback: Did Sean Connery Write an Angry Letter to Steve Jobs? | wafflesatnoon.com'}

Figure 3: Example of increasing the number of possible answers to a question to three. For each claim, three evidences are created that are the same as the following QA pairs.

Therefore, in addition to the AVeriTeC Score, we compared how well the four-class classification of final judgments was performed using Label Accuracy, ignoring the Evidence evaluation score. Since the Label Accuracy is expected to be higher when the necessary evidence for judgment is retrieved, it can be considered an indicator of how well the evidence retrieval was performed. Additionally, since no comparison with the correct Evidence is required, the problem with AVeriTeC Score, where useful information must be retrieved from sources other than the correct evidence, does not become an issue (though there is a possibility of accidentally making the correct judgment based on inappropriate evidence).

The experimental results are shown in Table 4. A comparison of the first and second rows of this table shows that the Label Accuracy for Question Top 1 is higher than the Label Accuracy for Claim Top 3. This suggests that with the current Evidence evaluation score, a small difference in Answer scores

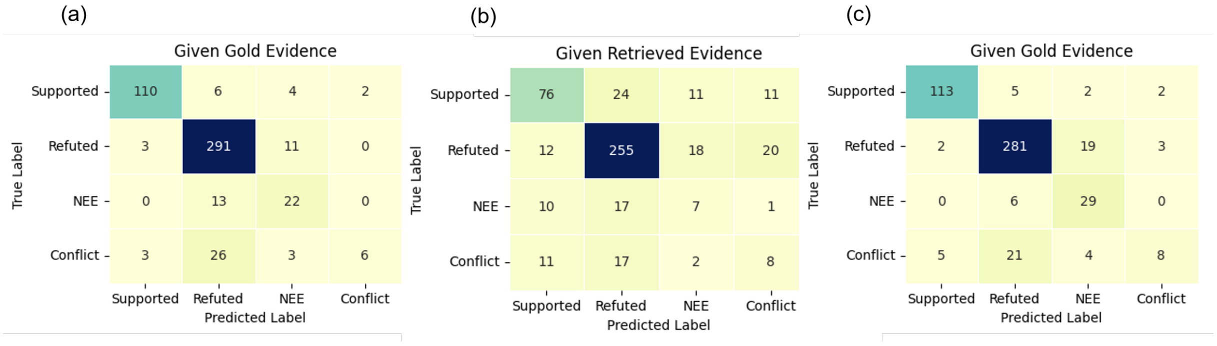


Figure 4: Confusion Matrix of verdict result of GPT-4o. (a) Given gold evidence with prompt Figure 2(d), (b) Given Retrieved evidence with prompt Figure 2(c), (c) Given gold evidence with removing “If there is even the slightest possibility that it is incorrect, output "Refuted"” from prompt Figure 2(d)

of around 0.1, as observed in Table 3, cannot be conclusively interpreted as a decline in retrieval performance.

To further improve the score,we considered the top 3 search results for each question (i.e., when a total of 9 sentences were retrieved). Then we included the Top 3 sentences as Evidence, noting the increase in URL hit rate (Table 2). However, if there are three answers for each question, each question will be reused three times. In this case, if an appropriate question can be created, there is concern that the evidence evaluation score may be unfairly high. Therefore, as shown in Figure 3, we used only the QA pair for the Top 1 answer, leaving the Question field empty for the Top 2 and Top 3 answers, and including them as evidence. In competitions using this dataset, participants can use up to 10 QA pairs. By following this limitation, we select the Top 3 answer sentence. This approach allowed for a fair evaluation of the AVeriTeC Score. The prompt given to GPT-4o in this case is shown in Figure 2(c). The judgment results are shown in the third row of Table 4, where both the Evidence score and judgment score improved by considering more Evidence.

Based on these results on validation dataset, the final form of the system was determined to involve searching based on RAG-Fusion, including three candidate answers in the questions, and making the final judgment using GPT-4o. The scores on the test data were Q 0.3774, Q+A 0.2851, and AVeriTeC Score 0.3865, with a rank of 8 on the leader-board.

4 Error Analysis

Figure 4(a)(b) shows the confusion matrix when the correct data or retrieved data using a RAG-Fusion-based search is provided. It can be seen that when the correct label is “NEE (Not Enough Evidence)” or “Conflict”, there is a tendency to predict it as “Refuted”. This is likely due to the instruction included in the prompt: “If there is even the slightest possibility that it is incorrect, output ’Refuted’.” However, in Fact-checking, to accurately predict “Refuted” claims as Refuted is the most important. Since it is crucial not to provide the user with incorrect information, it is undesirable to remove this instruction from the prompt.

Figure 4(c) shows the confusion matrix when this instruction is removed and the correct evidence is provided, revealing an increased risk of failing to detect Refuted claims, even when the information is complete.

To address this, adopting the concept of Corrective Retrieval Augmented Generation (CRAG) (Yan et al., 2024) could be considered for “NEE”. In CRAG, a new module is introduced to determine whether the retrieved document is necessary or not. If we incorporate the module into our system, we could first determine whether the information is enough or not. If the information is not enough, the system would classify it as “NEE”. If the information is enough, the system would proceed to classify the remaining three classes using the similar prompt as in 2(d). By adopting this new module, we will be able to improve the performance of “NEE”.

As a test, using GPT-4o, we performed a two-class classification—whether the information was complete—using the prompt from Figure 2(e) with the correct data provided. In this task, “Supported”, “Refuted”, and “Conflict” were considered as having complete information, while “NEE” was considered as lacking information. The accuracy rates were 90% for “Supported”, 86% for “Refuted”, 60% for “NEE”, and 78% for “Conflict”. Therefore, further prompt improvements are needed to adapt GPT-4o to this two-stage approach. Finetuning BERT should also be considered.

The “Conflict” class is difficult to render a verdict on, so further improvements will be necessary.

5 Another Approach

In this section, we will introduce a classification approach that we experimented with but did not yield satisfactory results. Although the performance did not exceed that of GPT-4o’s 4-class classification, we will present it here in the hope that it may contribute to future efforts by other participants.

We considered fine-tuning BERT as the final classifier for 4-class classification. However, the dataset exhibits a bias in the classification labels (in the training data: “Supported” 27.6%, “Refuted” 56.8%, “Not Enough Evidence (NEE)” 6.4%, “Conflicting Evidence/Cherry-picking” 9.2%). In particular, the “NEE” and “Conflict” labels are underrepresented. To address this, we devised two separate classifiers: one for “Supported” and another for “Refuted”. These classifiers perform binary classification, with the Supported classifier determining whether a claim is “Supported” or not, and the “Refuted” classifier determining whether a claim is “Refuted” or not. The final prediction label for the claim is then determined based on the results of these classifiers.

If the Supported classifier predicts True and the Refuted classifier predicts False, the final prediction is “Supported”. Conversely, if the Supported classifier predicts False and the Refuted classifier predicts True, the final decision is “Refuted”. If both classifiers predict False, the decision is “NEE”, and if both predict True, it is “Conflict”. This approach can mitigate the issue of label imbalance. For example, in the Supported classifier, claims that are annotated as “Supported” are used as positive examples, while “Refuted” and “NEE” claims are used as negative examples. This allows for similar treatment of “Refuted” and “NEE” labels.

We fine-tuned bert-base-uncased[5] for both a 4-class classifier and the combined two-classifier approach (batch size=32, learning rate=1e-5, with the training data split 9:1 and used for fine-tuning). The label accuracy on the validation data, when provided with correct evidence, was 0.536 for the 4-class classifier and 0.60 for the combined two-classifier approach. These results indicate that combining the two classifiers yields higher accuracy. However, as shown in the fourth row of Table 4, simply using GPT-4o for 4-class classification achieves a sufficiently high accuracy of 0.858, so this approach was not adopted for our system. We also conducted experiments where GPT-4o was assigned the task of the two classifiers, but the Refuted classifier did not perform well. We believe the issue arises because the difference between being “Refuted” and lacking the evidence to determine if it is “Refuted” has become unclear.

6 Conclusion

This paper discusses a method for solving the AVeriTeC Task. The proposed system, inspired by RAG Fusion, pre-generates questions for information retrieval. This approach allows for a greater amount of information to be used in searches compared to using only the claims. The Label Accuracy and AVeriTec Score showed that pre-generating questions resulted in higher accuracy.

Proposing an evaluation metric that can consider information beyond the currently accepted evidence when making judgments may lead to more appropriate progress in future research and development. Given the rapid advancement of LLMs, there is also a need to conduct research on adopting LLMs for the evaluation of evidence validity.

Limitation

In this system, the search for answers to questions is conducted using embedding vectors. This approach carries the risk of reducing the validity of the Question-Answer pairs compared to the method where the relevant sentences are searched first and the question is generated afterward. However, as shown in Table 4 of the current dataset, the approach of generating the question first and then searching for the answer yields higher accuracy, indicating that the validity of the Question-Answer pairs has not been compromised. Nonetheless, when the search for answers is more challenging, such as in highly specialized domains like medicine or biology, it is necessary to carefully verify the validity of the QA pairs.

While the current system primarily uses GPT-4o, further experiments with other models are necessary to verify its generalizability.

[5] https://huggingface.co/google-bert/bert-base-uncased

Acknowledgments

This work was supported in part by JSPS KAKENHI (Grant No. JP23K20733).

References

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.

Mitchell DeHaven and Stephen Scott. 2023. BEVERS: A general, simple, and performant framework for automatic fact verification. In *Proceedings of the Sixth Fact Extraction and VERification Workshop (FEVER)*, pages 58–65, Dubrovnik, Croatia. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

In-Zu Gi, Ting-Yu Fang, and Richard Tzong-Han Tsai. 2021. Verdict inference with claim and retrieved elements using RoBERTa. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 60–65, Dominican Republic. 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.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023. FactKG: Fact verification via reasoning on knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16190–16206, Toronto, Canada. Association for Computational Linguistics.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4551–4558. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Zackary Rackauckas. 2024. Rag-fusion: A new take on retrieval augmented generation. *International Journal on Natural Language Computing*, 13(1):37–47.

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

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*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

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.

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.

Jianshu Chen Yunkai Zhang Hong Wang Shiyang Li Xiyou Zhou Wenhu Chen, Hongmin Wang and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *Preprint*, arXiv:2401.15884.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.