UHH at AVeriTeC: RAG for Fact-Checking with Real-World Claims

Özge Sevgili, Irina Nikishina, Seid Muhie Yimam, Martin Semmann, Chris Biemann

Language Technology Group, Dept. of Informatics &
Hub of Computing and Data Science,
Universität Hamburg
Germany
{oezge.sevgili.ergueven,irina.nikishina,seid.muhie.yimam,
martin.semmann,chris.biemann}@uni-hamburg.de

Abstract

This paper presents UHH's approach developed for the AVeriTeC shared task. The goal of the challenge is to verify given real-world claims with evidences from the Web. In this shared task, we investigate a Retrieval-Augmented Generation (RAG) model, which mainly contains retrieval, generation, and augmentation components. We start with the selection of the top 10k evidences via BM25 scores, and continue with two approaches to retrieve the most similar evidences: (1) to retrieve top 10 evidences through vector similarity, generate questions for them, and rerank them or (2) to generate questions for the claim and retrieve the most similar evidence, again, through vector similarity. After retrieving the top evidences, a Large Language Model (LLM) is prompted using the claim along with either all evidences or individual evidence to predict the label. Our system submission, **UHH**, using the first approach and individual evidence prompts, ranks 6th out of 23 systems.

1 Introduction

Fact-checking is a process to (automatically) assess the truthfulness of a claim, which is an important task for some domains, e.g. journalism (Guo et al., 2022; Thorne et al., 2018; Thorne and Vlachos, 2018; Vlachos and Riedel, 2014). The AVeriTeC shared task[1](Schlichtkrull et al., 2023) aims at dealing with the challenge of verifying real-world claims with pieces of evidence from the Web, as shown in Figure 1.

Recently, Retrieval-Augmented Generation (RAG) provides a remedy for some issues of Large Language Models (LLMs), e.g. hallucination, while increasing the performance of especially knowledge-intensive tasks, including fact-checking (Gao et al., 2024). Motivated by this, we investigate how to effectively leverage such a method in this shared task.

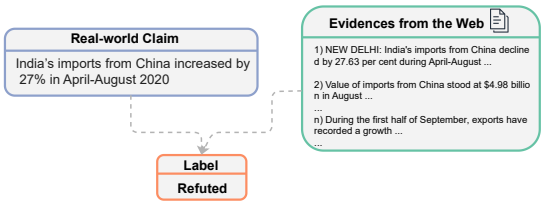

Figure 1: An example claim and several example evidences for this claim provided by organizers.

Our submission's pipeline is as follows; evidences (in the form of short texts like sentences[2]) per claim provided by task organizers are ranked using BM25 (Robertson and Zaragoza, 2009) and the top 10k evidences are selected. For retrieving the most relevant evidences, we consider two approaches: (1) **Retrieve-Question:** retrieving the most similar 10 evidences using vector similarity and generating questions for these evidences. Then, evidences are reranked again based on vector similarity with evidences in the form of question-answer.; (2) **Question-Retrieve:** generating questions for a claim, inspired from Chen et al. (2022), where they see an improvement for the retrieval with decomposed questions. We retrieve the single-best evidence per a question using vector similarity. The two approaches perform competitively in the development set. In the last step, we prompt LLM with the retrieved evidences to predict the label. We experiment to prompt with either all evidences or one evidence at a time. In our experiments, prompting with individual evidence can reach higher scores. Note that our pipeline resembles the steps conducted in the organizer's baseline (Schlichtkrull et al., 2023), especially in the Retrieve-Question approach, for more details see Section 4.

The contributions of this paper as follows:

- We investigate the use of RAG in the fact-

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

[2] Thus, we use evidence and sentence, interchangeably.

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 55–63

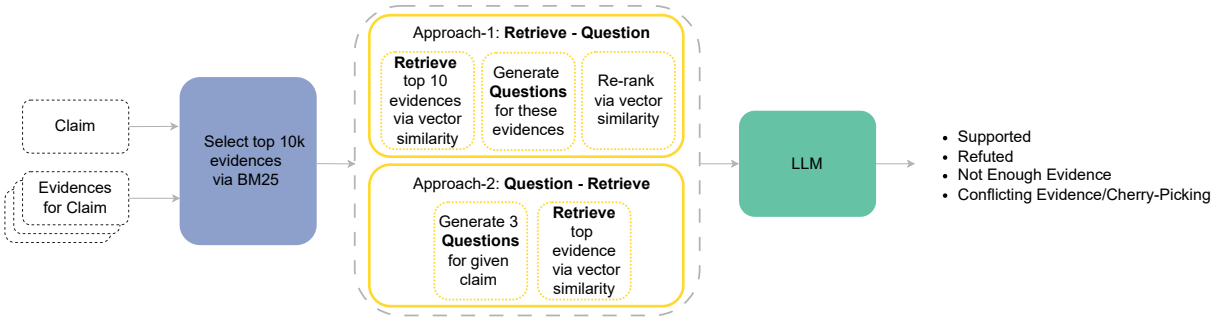

Figure 2: Inputs are the claim and evidences for this claim provided by task organizers. Top 10k evidences are selected with BM25 scores. Top question-sentence pairs are retrieved with Approach-1 (Retrieve-Question) or Approach-2 (Question-Retrieve). An output label is generated with LLM, prompted with either all pairs or individual pair.

checking task with real-world claims and evidences from the Web.

- We increase the baseline AVeriTeC score by more than three times, from 0.11 to 0.45, ranking 6th among 23 systems.

Considering the fact that our method is highly similar to the baseline, we also provide a list of main differences and/or improvements:

- We use top-10 evidences instead of top-3;
- We select 10K sentences with BM25 instead of 100 in baseline;
- Our Approach 2 is different than their pipeline;
- For veracity prediction, we rely on RAG-based predictions, i.e. incorporate evidence(s) into the prompt, while they use a finetuned BERT-large model.

Our code[3] is publicly available. The remainder of the paper is structured as follows. We continue with the background, and then the methodology is explained in detail. In subsequent sections, we present the experimental setup and discuss the results. And finally, conclusions, future work, and limitations are discussed.

2 Background

Retrieval-Augmented Generation LLMs have shown good performance on many tasks with their emergent abilities, e.g. in-context learning (Zhao et al., 2023). Yet, they still have some issues, e.g. hallucination. To resolve such issues, RAG integrates external information into LLMs (Fan et al., 2024; Gao et al., 2024; Li et al., 2024). Recently, many techniques have been developed for RAG in many aspects, for example, RaLLe (Hoshi et al., 2023) provides a framework for the evaluation of RAG approaches. Additionally, RAG has been applied to many tasks, e.g. question answering, fact checking, etc. We refer the readers to surveys, e.g. by Fan et al. (2024); Gao et al. (2024), for more information.

Fact-Checking It is a challenging task to automate a fact-checking process (Guo et al., 2022; Thorne and Vlachos, 2018), with different issues, for example, Chen et al. (2022) discuss the challenges of complex political claims. Many datasets have been developed for this task, e.g. the FEVER (Thorne et al., 2018) dataset from Wikipedia sources. In the AVeriTeC shared task, the dataset contains real-world claims, as shown in Figure 1, annotated with question-answer pairs.

3 Methodology

Overview The pipeline used in our solution is shown in Figure 2. Evidences per claim provided by task organizers are first ranked using BM25 (Robertson and Zaragoza, 2009). The highest-ranked 10k evidences to an input claim are selected. We have experimented two approaches to select the most similar evidences: (1) retrieving top 10 evidences first and then generating questions from evidences (Retrieve-Question), or (2) generating questions for a claim and retrieving the most similar evidence per question (Question-Retrieve). After the most similar evidences to a claim are retrieved, they are used to prompt LLM together with a claim.

[3] https://github.com/uhh-hcds/UHH-at-AVeriTeC

```
From the sentence below, please formulate 1 question that could be answered with this question. This question and answer should help to do the fact checking for the claim that is also given. Which question would be asked to get this asnwer given that we need to know whether the claim is true?
Examples:
claim: ...
answer: ...
question: ...
...
```

Figure 3: Prompt for Retrieve-Question Approach

Based on an LLM response, one of the labels, `Supported`, `Refuted`, `Not Enough Evidence`, `Conflicting Evidence/Cherrypicking`, is assigned.

3.1 Selecting Evidences via BM25

The task organizers provide a document collection in the form of short text for each claim. First, we make all sentences unique by keeping url references, to reduce the computation time and keep provenance. We apply BM25 to rank these evidences per claim. Then, the top 10K closest evidences to a given claim are selected.

3.2 Approach-1: Retrieve-Question

In this approach, vector representations for a claim and 10k sentences are created. Vector similarities between each sentence and claim are computed. The most similar 10 sentences to a claim are retrieved. Next, we generate a question using LLM for each of these top 10 sentences with the prompt, which is shown in Figure 3.

The vector representations for question + answer and claim are created. Evidences are reranked based on similarity of claim and each evidence in the form of question and answer. We experiment with {3,5,7,10} evidences for the next step.

3.3 Approach-2: Question-Retrieve

First, 3 questions are generated for each claim using the prompt in Figure 4. For each question, the most similar sentence is selected using the similarity between vectors of 10K sentences and the question and claim vector.

```
From the sentence below, please formulate up to 3 questions to help to do the fact-checking. What do we need to know to check whether the claim is true? "Decompose" the claim into subquestions. Generate as few questions as possible.
Example:
claim: ...
questions: ...
...
```

Figure 4: Prompt for Question-Retrieve Approach

3.4 LLM Strategies

In the typical RAG (Gao et al., 2024), all selected documents and claims are combined into a prompt. We experiment two ways, either as in the common RAG or to utilize one retrieved document at a time and then based on individual predictions, assign one label, inspired from the baseline (Schlichtkrull et al., 2023). The prompt[4] that we use in our experiments for the first alternative is shown in Figure 5.

```
<s>[INST]
Classify the claim into “Supported”, “Refuted”, “Not Enough Evidence”, or “Conflicting Evidence/Cherrypicking” based on list of evidences.
No Explanation, No Note! Your respond should be in JSON format containing `“label”` key-value pair without any further information. For instance,
```json
{
“label”: “Supported”
}
```
User Claim: ...
Evidences: [...]
Class: [/INST]
```

Figure 5: Prompt for a label with all evidences

The prompt for the second option also includes a prediction of a score, as shown in Figure 6. The score prediction is only used to assign a label `Not Enough Evidence`. If LLM has no pre-

[4]We use as a reference: https://www.pinecone.io/learn/mixtral-8x7b/

diction of Refuted or Supported (or it generates something different or more), and the score is smaller than or equal to 0.5, then Not Enough Evidence is assigned. Therefore, a smaller score is used for the Not Enough Evidence label. We have two strategies to assign a final label from individual evidence labels. In the first one, similar to the baseline, if all labels from evidences are the same, this label is assigned, otherwise Conflicting Evidence/Cherrypicking. In the other one, again if there is only one label, the predicted label will be assigned; if there are only two different labels from evidences, then the majority is assigned. Otherwise, Conflicting Evidence/Cherrypicking is assigned.

LLM might generate different texts than only the label output, in these cases, we assign Refuted, as it is the most common label in the training set[5].

```
<s>[INST]
Classify the claim into “Supported” or
“Refuted” based on list of evidences.
Produce a score for the class label.
No Explanation, No Note! Your respond
should be in JSON format containing
`“label”` key-value pair without any
further information. For instance,
```json
{
“label”: “Supported”
“score”: 0.7
}
```
User Claim: ...
Evidence: ...
Class: [/INST]
```

Figure 6: Prompt for a label with individual evidence

4 Experimental Setup

4.1 Data, Evaluation, and Baseline

Data The task organizers provide real-world claim files for training, development, and testing that contain 3068, 500, 2215 samples, respectively. They also provide document collections for each claim from the Web, and we leverage these given document collections.

Evaluation Evaluation is done by organizers and based on the agreement between predicted evidences and gold ones with the scoring function of METEOR (Banerjee and Lavie, 2005), computing for question-only pairs (Q) or question and answer pairs (Q+A). If this evidence score is higher than a cutoff value of 0.25, then veracity predictions are evaluated, referred to as Veracity@25 or AVeriTeC score, in this paper. For more information, we refer to the paper by Schlichtkrull et al. (2023).

Baseline The pipeline in the baseline, provided by organizers, starts with collecting evidences from the Web by searching via Google Search API for each claim. Our Retrieve-Question approach pipeline is similar to their pipeline. For example, the next step in the AVeriTeC approach is to filter top 100 sentences using BM25, and then to generate a question for each sentence using BLOOM (Workshop et al., 2023). The question-answer pairs are reranked with a fine-tuned BERT-large model (Devlin et al., 2019). The number of top evidences and models differ in our experiments. For final step of the veracity prediction, AVeriTeC leverages a fine-tuned BERT-large model for an individual question-answer pair prediction with a label of supporting, refuting, or irrelevant. If all labels are Supported or Refuted, the respective one is assigned, else if there are both labels, Conflicting Evidence/CherryPicking is assigned. If no label is assigned based on these two conditions, then Not Enough Evidence is assigned. Our LLM strategy with individual prompt (Figure 6) along with the first strategy is similar to their veracity prediction.

4.2 Implementation Details

For the computation of vectors, we use the model Alibaba-NLP/gte-base-en-v1.5[6] (Li et al., 2023; Zhang et al., 2024), which is available in Hugging Face (Wolf et al., 2020), using sentence-transformers (Reimers and Gurevych, 2019). We choose this model from Hugging Face's MTEB leaderboard[7] by using the "Retrieval" task and the "FEVER" data, as we consider this task and data are relevant to the shared task. This model was ranked 2nd in the leaderboard[8];

[5] For the best model with unique sentences (with veracity score, 0.40) in Table 1, we assigned Refuted for 1573 evidences over 5000 evidences, while for test submission 6767 over 22150 evidences were assigned Refuted.

[6] https://huggingface.co/Alibaba-NLP/gte-base-en-v1.5

[7] https://huggingface.co/spaces/mteb/leaderboard

[8] checked on a date - 08.07.2024

LLM	Retrieval Approach	LLM prompt	top-n	unique sentences	Q	Q+A	Veracity@0.25
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Question-Retrieve	1	3	✔	0.37	0.24	0.19
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	1	3	✔	0.40	0.24	0.19
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	1	5	✔	0.44	0.27	0.23
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	1	7	✔	0.46	0.28	0.27
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	1	10	✔	0.48	0.30	0.30
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	2-1	10	✔	0.48	0.30	0.19
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	2-2	10	✔	**0.48**	**0.30**	**0.40**
Mixtral-8x7B-Instruct-v0.1 (quantized 4bit)	Retrieve-Question	2-2	10	✘	**0.49**	**0.31**	**0.42**
Meta-Llama-3.1-8B-Instruct (quantized 4bit)	Retrieve-Question	2-2	10	✔	0.48	0.30	0.26
GPT-4o-mini	Retrieve-Question	2-2	10	✔	0.48	0.30	0.38
Baseline					0.24	0.19	0.09

Table 1: Results of different approaches on the development for Q, Q+A, Veracity@0.25 scores are shown. Baseline is provided by task organizers. **LLM:** name of LLM model, used in the generation step. **Retrieval Approach:** either Retrieve-Question (first retrieve sentences with vector similarity, generate questions for sentences, and rerank with vector similarity, including questions) or Question-Retrieve (generate questions for a claim and retrieve a sentence based on vector similarity, including questions). **LLM prompt:** either all evidences at once (1) or one by one (2) - (2-1, 2-2) used strategy 1 or 2 for a final label assignment. **top-n:** number of evidences used for the prompt. **unique sentence:** either to make sentences unique before BM25 or not.

however, we preferred it over the first-ranked model due to a lower dimension size of 768.

For question generation, we experiment with `GPT-4o-mini` LLM from OpenAI. For the LLM in the generation step, we have experimented with `mistralai/Mixtral-8x7B-Instruct-v0.1`, `Meta-Llama-3.1-8B-Instruct`[9] with 4-bit quantized, also available in Hugging Face and `GPT-4o-mini`. For BM25, we use the rank-25 library[10], as used in the baseline system, and we use the NLTK library (Bird et al., 2009) to tokenize claims and evidences.

5 Results

We report Q, Q+A, and Veracity@0.25 scores in Table 1, for the development set. According to the results, the veracity scores for Question-Retrieve and Retrieve-Question for the top 3 are the same, however, we continue with Retrieve-Question since the Q score is slightly higher. Although the difference is not that much, we continue with the higher one. Leveraging the top 10 evidences reaches best among top {3, 5, 7, 10} evidences. Prompting LLM with all evidences (LLM prompt 1 – Figure 5), is better than prompting individually with labeling strategy 1 (LLM prompt 2-1 – Figure 6), however, strategy - 2 (LLM prompt 2-2 – Figure 6) reaches higher score. As explained in Section 3.1, we make sentences unique to reduce the computation time, yet for the development set we have also experimented without applying this, as marked with a cross in the "unique sentence" field in Table 1 and observed an improvement. However, since the number of evidences is larger in the test set, we rather prefer to compute with unique sentences for efficiency. We also experiment with two different LLMs, namely `Meta-Llama-3.1-8B-Instruct` and `GPT-4o-mini` with the same prompt, the latter one is competitive with the `Mixtral-8x7B-Instruct-v0.1`.

[9] `https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1`, `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct` with pipeline parameters top_k=50 and repetition_penalty=1.204819277108434 by referencing Hoshi et al. (2023), and do_sample=False and max_new_tokens=32

[10] `https://pypi.org/project/rank-bm25/`

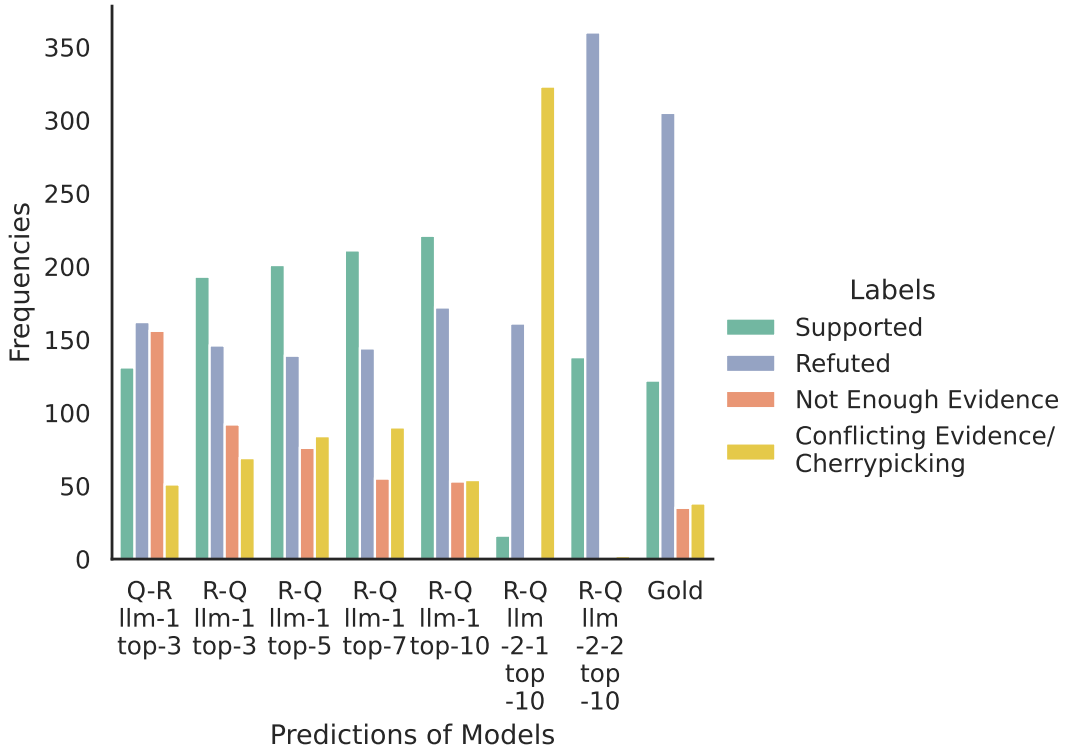

Figure 7: The frequencies of the predicted labels for different model configurations and gold labels on the development set are shown. **Q-R/R-Q:** Question-Retrieve or Retrieve-Question approach. Other configurations are the same as in Table 1.

Rank	Participant Team	Q	Q+A	AVeriTeC
1	TUDA_MAI	0.45	0.34	**0.63**
2	HUMANE	0.48	**0.35**	0.57
3	CTU AIC	0.46	0.32	0.50
4	Dunamu-ml	**0.49**	**0.35**	0.50
5	Papelo	0.44	0.30	0.48
6	**UHH**	0.48	0.32	0.45
20	Baseline	0.24	0.20	0.11

Table 2: Results of baseline and models ranked above our system, UHH, on the test computed and provided by task organizers for Q, Q+A, AVeriTeC scores are displayed.

Table 2 shows the test set results provided by task organizers. We display the systems results that ranked above us and the baseline scores, however in total there are 23 results in the leaderboard[11]. Our approach improves the baseline score and is ranked 6th. Our Q score is in the top 3 and the AVeriTeC score is more than quadrupled as compared to baseline.

5.1 Analysis

To analyze the results, we first built a class distribution of the predicted results with all our approaches and compared them with the gold standard label distribution. From Figure 7, we can see that the `Refuted` class has the highest frequency, making it the most common label. In contrast, all models tend to predict `Supported` or `Not Enough Evidence` labels more frequently than `Refuted`, leading to a significant mismatch between the models' predictions and the gold standard. For `Conflicting Evidence/Cherrypicking`, all models predict it

[11] https://eval.ai/web/challenges/challenge-page/2285/leaderboard/5655

Claim ID - Claim	Individual Predictions	Evidences	Final Prediction	Gold Label
217 - Nigeria's current population exceeds 200 million.	Refuted	**Q:** What is Nigeria's current estimated population? **A:** With a population of roughly 200 million people, Nigeria's	Supported	Supported
	Supported	**Q:** What is the current population estimate for Nigeria? **A:** Nigeria's population is projected to reach 262.9 and 401.3 million people in 2030 and 2050, respectively.		
	Refuted	**Q:** What is the current estimated population of Nigeria? **A:** The population of Nigeria is currently estimated at 198 million, with an annual		
	Supported	**Q:** What is Nigeria's estimated population in comparison to 200 million? **A:** With over 220 million people, Nigeria is the most populated country in Africa and the sixth in the world.		
	Refuted	**Q:** What is the estimated population of Nigeria? **A:** Nigeria has a population of 180 million people (seventh largest in the world) and an economy worth more than $500 billion (21st in the world).		
	Refuted	**Q:** Is Nigeria currently the most populous country in Africa? **A:** Nigeria is the most populous country in Africa and the eighth most populous country in the world, with approximately 162 million people.		
	Supported	**Q:** What was Nigeria's population in 2021? **A:** - The population of Nigeria in 2021 was 213,401,323, a 2.44% increase from 2020.		
	Supported	**Q:** What was Nigeria's population in 2022? **A:** - The population of Nigeria in 2022 was 218,541,212, a 2.41% increase from 2021.		
	Supported	**Q:** What was Nigeria's population in 2020? **A:** Nigeria had a population of 206.14 million people (2020) with an annual population growth rate of 2.5%.		
	Supported	**Q:** What was Nigeria's population as of 2008? **A:** Nigeria is a West African country with about 152 million people (as of 2008). It is by far		
327 - Carlos Gimenez approved a 67% pay raise for himself and increased his own pension.	Refuted	**Q:** Did Carlos Gimenez approve a pay raise for himself? **A:** The amount of money that employees are voluntarily putting into their own pension funds has more than doubled and 70% of employees say they've paid off debt.	Refuted	Supported
	Refuted	**Q:** Did Carlos Gimenez approve a pay raise for himself and increase his pension? **A:** to accrue benefits under the defined benefit pension arrangements, net of his own contributions.		
	Refuted	**Q:** What changes did Carlos Gimenez make to his pay and pension? **A:** subsequently increased the monthly pension rate above what had		
	Refuted	**Q:** Did Carlos Gimenez approve a pay raise for himself and increase his pension? **A:** Gimenez gets a pension of about $120,000 a year from the city of Miami, and has caught heat from labor for opposing the salary hikes for county employees.		
	Refuted	**Q:** What changes to retirement age and pension plans were approved under Carlos Gimenez? **A:** retirement age will gradually increase to 67 by the year 2027, and		
	Refuted	**Q:** What was Carlos Gimenez's salary before the pay raise? **A:** By jacking his own salary up $100,000 for the last two years to $250,000, he significantly improves that average.		
	Refuted	**Q:** What significant changes did Carlos Gimenez implement regarding pay and pensions upon taking office? **A:** huge boost when Carlos Gimenez came into the office		
	Supported	**Q:** What percentage of pay increase did Carlos Gimenez approve for himself? **A:** Read related: Termed out Mayor Carlos Gimenez gives self undeserved 70% pay raise		
	Supported	**Q:** Did Carlos Gimenez authorize a pay raise for himself while making budget cuts in Miami-Dade? **A:** In his time in office, Giménez gave himself a 67% pay raise, and kept a taxpayer funded Mercedes while cutting $400 million in Miami-Dade jobs and investment.		
	Refuted	**Q:** What actions did Carlos Gimenez take regarding pay raises and pensions during his tenure as mayor? **A:** Remember, former Mayor Carlos Alvarez gave big raises to his inner circle also before he was recalled so that Gimenez — or Carlos II, as some have taken to call him — could be elected.		
421 - The CDC recommended wearing only certain beard styles to help prevent the spread of coronoavirus.	Supported	**Q:** Did the CDC recommend wearing only certain beard styles to help prevent the spread of coronavirus? **A:** The CDC recommends shaving beards to protect against the virus	Refuted	Refuted
	Refuted	**Q:** What does the CDC say about beard styles in relation to preventing the spread of coronavirus? **A:** The CDC did not, and does not, recommend that men shave their beards to protect against the SARS-CoV-2 virus.		
	Refuted	**Q:** What does the CDC recommend regarding beard styles in relation to preventing the spread of coronavirus? **A:** To recap, CDC beard advice is not to shave your beard. Coronavirus prevention is best done by washing your hands and practicing social distancing while wearing a cloth face covering.		
	Refuted	**Q:** Did the CDC recommend specific beard styles for preventing the spread of coronavirus? **A:** It's advice about which beards block respirators. The CDC has not said anything about shaving beards for this Coronavirus.		
	Refuted	**Q:** Is the CDC recommending specific beard styles to prevent the spread of coronavirus? **A:** And while facial hair could interfere with respirator masks, the CDC has not recommended people shave their beards to ward off the virus.		
	Refuted	**Q:** What does the CDC say about beard styles and their impact on preventing the spread of coronavirus? **A:** A headline claims that the CDC recommends men shave their beards to protect against coronavirus.		
	Refuted	**Q:** Did the CDC issue guidelines regarding facial hair styles for preventing the spread of coronavirus? **A:** Social media users sharing a CDC infographic showing various styles of facial hair have suggested that the agency is instructing people to shave beards and mustaches to prevent the coronavirus.		
	Refuted	**Q:** What does the CDC say about facial hair styles in relation to the use of respirators? **A:** While the Centers for Disease Control and Prevention (CDC) recommends against certain facial hair stylings for workers who wear tight-fitting respirators, it has not recommended shaving as a precaution to prevent COVID-19.		
	Refuted	**Q:** What guidelines has the CDC provided regarding personal hygiene related to the spread of coronavirus? **A:** The CDC has touted basic personal hygiene like avoiding touching your face and washing your hands since the coronavirus outbreak started, and the same type of cleanliness can be applied to beards.		
	Supported	**Q:** What does the CDC recommend regarding beard styles for effective mask use? **A:** The CDC says to shave your beard into one of a few acceptable styles so you can ensure a snug fit for a mask, if needed.		

Table 3: Some examples on the development set, where we leverage the majority choices based on Retrieve-Question approach along with LLM 2-2, top-10 evidences from unique sentences, and with Mixtral model.

less frequently, aligning with its lower occurrence in the gold standard but still under-predicting it relative to the gold standard's distribution.

Our major concern of the pipeline is "majority voting". One of the hypothesis is that many of the lower-level evidences are unrelated to the claim, making it easier for the LLM to determine that this claim is `Refuted`. In this case, majority voting is also likely to be `Refuted`. To check this, we manually analyze some samples with a majority and demonstrate the examples of different cases in Table 3. For example, the claim "Nigeria's current population exceeds 200 million" has `Refuted` label predictions at the top of the list, however, due to the majority vote, the correct label `Supported` is selected. If we counted only top 5 evidence into account, the final answer could be either `Refuted` (majority vote) or `Conflicting` (both labels are presented, no evident winner). Regarding the second example, we can see that the claim was refuted due to the majority of the retrieved evidence being classified as refuted. However, the majority vote in this case led to an incorrect classification. Regarding the third example, we can see the majority class `Refuted` is coherent with the correct answer, even though the top 1 evidence is classified as `Supported`.

From these examples, we can see that the higher-ranked evidences' labels are not coherent with the golden labels always, the top-10 retrieved evidences provide either correct or incorrect labels regardless the lower-ranked arguments.

6 Conclusion

We have described our UHH system that is submitted to the AVeriTeC shared task. We have explored the use of RAG in this task and have used different LLMs in different steps, with a different number of evidences - top {3, 5, 7, 10}. Top 10 evidences using `Mixtral-8x7B-Instruct-v0.1` (quantized 4-bit) model by prompting individual evidence (strategy 2-2) in the Retrieve-Question approach are ranked 6th in the shared task. In future work, we would like to investigate using a vector database. We have used the evidences as provided by organizers, and we also plan to experiment with different granularity of texts from these evidences.

Limitations

For the creation of unique sentences before BM25 ranking, we used the "set" operation that might change the order of sentences and this might affect the reproducibility regarding the same order of sentences. Additionally, we leverage LLMs, and it could produce different responses every time that might affect the results if reproducing the approach from scratch. However, we have saved the predictions that are used for the task submission. Thus, these predictions can be used to reproduce the results. It is important to note that the computation time for the LLM when predicting a label using strategy 2 is longer than that for strategy 1, as strategy 2 involves prompting individually for each piece of evidence.

Acknowledgements

This research was funded by the "Hamburgische Investitions- und Förderbank" in the project FaktenFassenKI.

References

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.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. *Preprint*, arXiv:2405.06211.

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.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Yasuto Hoshi, Daisuke Miyashita, Youyang Ng, Kento Tatsuno, Yasuhiro Morioka, Osamu Torii, and Jun Deguchi. 2023. RaLLe: A framework for developing and evaluating retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 52–69, Singapore. Association for Computational Linguistics.

Mahei Manhai Li, Irina Nikishina, Özge Sevgili, and Martin Semmann. 2024. Wiping out the limitations of large language models – a taxonomy for retrieval augmented generation. *Preprint*, arXiv:2408.02854.

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

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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, 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.

James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico. 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.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and et. al. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *Preprint*, arXiv:2211.05100.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. *Preprint*, arXiv:2407.19669.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *Preprint*, arXiv:2303.18223.