Misinformation Detection with Question-Answering and RAG

Yeaeun Kwon
University of Arizona
yeaeunkwon@arizona.edu

Abstract

The proliferation of misinformation poses substantial risks to both society and online communities. In response, researchers have intensively studied misinformation detection by training a model on specific data. However, relying solely on Large Language Models (LLMs) has limitations, particularly when verifying complex or novel claims that exceed the models' internal knowledge. This project addresses these limitations by incorporating external knowledge for misinformation detection. Additionally, each claim is decomposed into two questions to identify the specific information necessary for verification. Results demonstrate that the LLaMA model achieves significant improvements in fact-verification, whereas GPT models perform effectively with document-based evidence alone to verify claims.

1 Introduction

Anonymity on social media has promoted the spread of misinformation with reducing individual's accountability (Del Vicario et al., 2016). The harmful effect of misinformation on society, extending beyond online communities, has underscored the need for effective detection methods. This need became even more critical during the pandemic, when misinformation surged and, at times, posed significant health risks to people (Roozenbeek et al., 2020).

In response, researchers have extensively studied misinformation detection. Earlier studies primarily utilized synthesized datasets and pre-trained models, leveraging these datasets as inputs for classification (Lee et al., 2020; Yue et al., 2021). However, more recent research has expanded to explore real-world data from sources such as social media and political contexts. With advancements in large language models (LLMs), methods that combine the internal knowledge of LLMs with external knowledge from other sources have demonstrated significant improvements in misinformation detection performance(Wan et al., 2024; Zhang et al., 2024; Khaliq et al., 2024; Yue et al., 2024a,b).

In this project, I utilize prompting strategies and the retrieval augmented generation(RAG) technique to detect misinformation. Beyond leveraging the internal knowledge of LLMs, I incorporate evidence retrieved from external search engine with the RAG technique. This evidence serves as answers to questions derived from decomposing each claim in the FEVER dataset (Thorne et al., 2018). Decomposing claims into questions helps clarify the specific information needed to verify them, and utilizing external knowledge enables LLMs to detect misinformation by incorporating newly available or evolving information, reflecting advances in knowledge.

2 Related Works

Misinformation Detection

In misinformation detection tasks, pre-trained language models (PLMs) have been widely utilized in previous research. Most of these models are trained using input content to perform classification. For instance, Lee et al. (2020) directly utilize the internal knowledge implicitly stored within PLMs' parameters for fact verification. In addition, Yue et al. (2021) leverage pseudo labeling to generate high-confidence target examples for joint training with source data. Beyond textual misinformation, the detection of multi-modal misinformation has also garnered significant attention due to the increasing prevalence of fake news involving images and videos. The multi-modal detection can also improve the performance. For example, Zhou et al. (2022) introduce a modality-wise attention module to adaptively reweight and aggregate the image and text features for fake news detection. However, models trained solely on input content often perform poorly when encountering unseen data. As

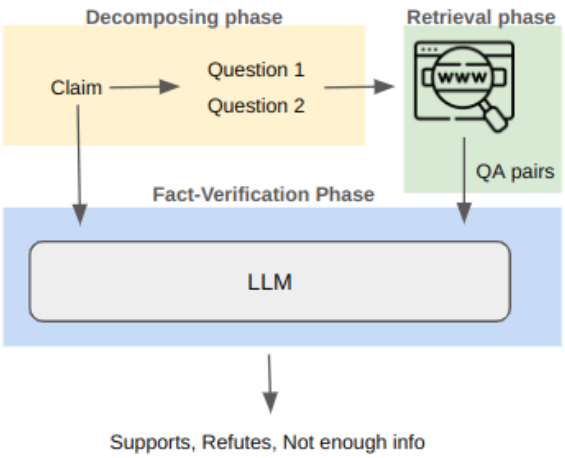

Figure 1: The overall system for misinformation detection. The system is structured into three phases: Decomposing phase, Retrieval Phase, and Fact-Verification Phase.

a result, evidence-based detection methods, which leverage external knowledge sources, have been developed. Particularly, Retrieval Augmented Generation(RAG), which is a technique that combines the ability of LLMs with information retrieval techniques, achieved notable performance in misinformation detection, generating evidence and explanations. (Wan et al., 2024; Zhang et al., 2024; Khaliq et al., 2024; Yue et al., 2024a,b)

In this project, I aim to detect misinformation by decomposing a claim into two sub-questions. These questions are then answered based on external documents retrieved using the RAG technique, giving more reliable evidence to support or refute the claim. This approach is implemented through sophisticated prompting and text similarity measures.

3 System

The misinformation detection system in this project is structured into three phases: the Decomposing phase, the Retrieval Phase, and the Fact-Verification Phase. In the Decomposing phase, an unverified claim is broken down into two questions that need to be answered to verify the claim. The Retrieval phase involves searching for relevant documents using a search engine API and apply text similarity methods to identify the relevant document to the questions. Lastly, in the Fact-Verification phase, LLMs generate one of three labels, *supports, refutes, and not enough info*, based on the claim and the generated question-answer pairs. The question-answer pairs play a role of evidence in the verification process. Figure 1 illustrates the overall framework of this system.

BertScore	ROU1
0.889	0.396

Table 1: The BerScore and ROUGE score of the generated questions. They are compared with the original claim.

3.1 Decomposing Phase

To determine the evidence required for verifying a claim, the claim is decomposed into two specific questions. This decomposition process clarifies what evidence needed for a claim and the reasoning behind why it may be considered misinformation, thus providing more explainable answers as evidence. To assess the validity of the generated questions, they are evaluated with ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020) with a claim, as detailed in Table 1. The generation of these questions is conducted through prompting to Gpt-3.5-turbo. The prompt is detailed in Appendix A.

3.2 Retrieval Phase

In the Retrieval Phase, external knowledge is incorporated into the answers to the generated questions. These questions are used as queries to search for relevant documents using the DuckDuckGo API[1]. Both the retrieved documents and the questions are embedded using E5 (Wang et al., 2024), a text embedding model trained through a contrastive learning approach. To find the most relevant documents for each query, cosine similarity between the embedded representations of the documents and questions is then calculated. Then, the top-k documents (with k=2 in this experiment) are selected to extract answers to the questions. LLMs subsequently generate answers to these questions based on the retrieved documents through the use of prompts. The detailed prompt is provided in Appendix A.

3.3 Fact-Verification Phase

LLMs are provided with the generated question-answer pairs and the original claim, and required to predict whether the question-answer pairs are support, refute, or provide insufficient information to verify the claim. LLMs assess the factuality

[1] https://pypi.org/project/duckduckgo-search/

of the claim based on these question-answer pairs, which serve as evidence for the verification process.

4 Experiment

4.1 Dataset

In this project, the FEVER dataset (Thorne et al., 2018) is utilized to detect misinformation. This dataset consists of manually synthesized claims derived from Wikipedia, with three possible labels: supports, refutes, and not enough information. However, given the development in knowledge since the dataset's creation, I exclude instances labeled as "not enough information" from the training and evaluation process, focusing only on claims labeled as "supports" or "refutes." In the Fact-Verification phase, the "not enough information" label is included as an option for the LLMs during prediction to account for uncertainty. A subset of the dataset is used, comprising 474 instances labeled as "supports" and 465 labeled as "refutes." Additionally, duplicates in the dataset were removed, ensuring that only unique claims were included in the analysis.

4.2 Baselines

Baseline models verify claims without decomposing them into sub-questions. Instead, the text of the original documents retrieved through the RAG method serve directly as evidence, without further processing into specific answers to questions. In this apporoach, the original claim is used as a query to search for relevant documents during the Retrieval phase, rather than using the questions. LLMs are provided with the claim and the top k documents, identified through the text embedding and cosine similarity. Based on these documents, the models generate one of the verification labels. For the baseline experiments, GPT-3.5-turbo and LLaMA3 (`Llama-3.2-1B-Instruct` from the HuggingFace library) are utilized.

4.3 Experimental models

The experimental models are provided with the original claim and corresponding question-answer pairs and are tasked with generate a rating, which should be one of the labels: supports, refutes, or not enough info. Although instances with the "not enough information" label from the FEVER dataset were excluded from this experiment, the model retains the option to predict this label to account for uncertainty.

4.3.1 LLM-QA

In this approach, questions are answered by LLMs based on a claim and the top k relevant documents from an external search engine (with k=2 for each question). The questions and the generated answers are presented as evidence in the fact-verification process. The prompt for generating answers is detailed in Appendix A.

4.3.2 LLM-Topk

each question is answered by retrieving the most similar five sentences from the top k relevant documents (with k=2 for each question). Cosine similarity is calculated between the claim and sentences within these documents, and only the top five sentences are selected as the response to each question. Gpt-3.5-turbo and LLaMA3 were tested, with the temperature is set to 0.1 to minimize result variability.

5 Results

1) Gpt-QA model performance relative to baseline

The Gpt-QA model does not demostrate a statistically significant improvement over the baseline model. While the Gpt-QA model shows a slightly higher accuracy, the p-value (p =0.45 =alpha/n with alpha=0.05 and n=2, the number of independent tests) suggests no significant difference from the baseline model. Additionally, the GPT models exhibit a high recall for the "supports" label and high precision for "refutes," indicating a tendency to predict "refutes" as "not enough info" more frequently than they predict "supports" (see Table 2). By contrast, the GPT-Topk model performs significantly worse than the baseline, with a Bonferroni-corrected p-value of 1.

2) Question-Answering Enhances LLaMA's Fact-Verification Performance

For LLaMA models, incorporating question-answer pairs as evidence substantially improves fact-verification performance over the baseline, yielding a Bonferroni-corrected p-value of 0 across two independent tests. Notably, the LLaMA-Topk model achieves the highest accuracy among the LLaMA models tested in this project. The differing performance between the GPT and LLaMA models could be due to the fact that the GPT models are accessed via an API, whereas the LLaMA models are

Model		P	R	F	A
GPT-base	Supports	0.74	0.91	**0.82**	0.725
	Refutes	0.92	0.52	0.67	
Gpt-QA	Supports	0.76	0.85	0.8	**0.727**
	Refutes	0.89	0.6	**0.71**	
Gpt-Topk	Supports	0.77	0.82	0.79	0.60
	Refutes	0.91	0.38	0.53	
LLaMA-base	Supports	0.59	0.55	0.57	0.437
	Refutes	0.55	0.31	0.4	
LLaMA-QA	Supports	0.67	0.57	0.62	0.569
	Refutes	0.59	0.56	**0.57**	
LLaMA-Topk	Supports	0.77	0.82	**0.79**	**0.60**
	Refutes	0.91	0.38	0.53	

Table 2: The results of the fact-verification experiments with P,R,F, and A representing precision, recall, f1-score, and accuracy, respectively.

Model		# of Not enough info
GPT-base	Supports	18
	Refutes	68
Gpt-QA	Supports	38
	Refutes	65
Gpt-Topk	Supports	65
	Refutes	174
LLaMA-base	Supports	43
	Refutes	95
LLaMA-QA	Supports	16
	Refutes	61
LLaMA-Topk	Supports	65
	Refutes	174

Table 3: The number of predicted "not enough info" label in the experiments. For example, GPT-base model predicted 18 instacnes with "supports" label as "not enough info".

static, downloaded from the HuggingFace library with pre-saved parameters.

3) High "Not Enough Information" Predictions for "Refutes" Instances

As presented in Table 2, the instances with “refutes” are more likely to be predicted as “not enough info” label across all models. The difference is notable even after considering the differences in the number of instances in the dataset. An error analysis is conducted in the following section to further investigate this tendency.

6 Error analaysis

To analysis errors, the predictions labeled as “not enough info” were sampled from each model, and the errors were categorized into four types: *Incorrect retrieval, Failed inference, Incorrect question generation, and Ambiguous answer*. The frequency of these errors varied across models.

1) Baseline Models with Relevant Documents

The baseline models were provided only with the top two relevant documents retrieved from the external search engine, without question-answer pairs. As a result, these models commonly presents errors related to incorrect retrieval and failed inference. Specifically, some retrieved documents were either irrelevant or insufficiently relevant to determine the correct label. In other cases, despite the documents serving as suitable evidence, the models failed to interpret the information, leading to misclassifications.

2) Models with Generated Answers Based on Relevant Documents

The models with the generated answers based on relevant documents commonly have errors related to failed inference and incorrect question generation. Even when questions were accurately generated and answers were reasonably clear, these models sometimes failed to assign the correct label for claim verification. Additionally, the incorrect question generation often led to the misclassification because the ambiguous questions generate incorrect and irrelevant answers, hindering the accuracy of fact-verification.

3) Models with Top-k Most Similar Sentences from Relevant Documents

The models that utilized the top k most similar sentences from relevant documents of each question frequently demonstrated errors related to ambiguous answers. Although the selected sentences closely matched the claim and questions, they often failed to provide sufficient evidence for accurate fact-verification in most error cases.

7 Conclusion

This project explores misinformation detection using question-answering and the Retrieval-Augmented Generation (RAG) technique, structured in three main phases. In the Decomposition Phase, claims are broken down into two specific questions through a prompting strategy. Next, in the Retrieval Phase, relevant documents are retrieved from the external DuckDuckGo API and are used as evidence of each claim for baseline models. For experimental models, answers are generated

from these documents either through prompting or by extracting the top five most similar sentences to each claim and question. Finally, in the Fact-Verification Phase, the claim is verified based on the generated answers and the relevant documents. Results show that for the GPT-3.5-turbo model, neither the GPT-QA nor GPT-Topk models exhibit significant improvement in fact-verification. However, for the LLaMA models, the LLaMA-QA and LLaMA-Topk models demonstrate significantly enhanced performance over the baseline.

This project has several limitations. First, only the top five sentences were extracted from each document retrieved from the search engine, but results may vary with changes in k. While the Gpt-Topk model shows the lowest performance among the experimental models, its accuracy could potentially improve with a higher k value. Second, the absence of in-context learning may contribute to instances of "refutes" being misclassified as "not enough info." Error analysis revealed that some instances labeled as "not enough info" actually contained sufficient evidence but were incorrectly classified, suggesting that in-context learning could enhance prediction accuracy. Finally, the FEVER dataset, a synthesized dataset based on Wikipedia, was used for this project, which may overlap with the data used training the large language models. Consequently, results may vary on real-world data, such as political information, which is more ambiguous and requires social-context. These limitations will be addressed in future studies and projects.

References

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the national academy of Sciences*, 113(3):554–559.

M. Abdul Khaliq, P. Chang, M. Ma, B. Pflugfelder, and F. Miletić. 2024. Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models.

Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language models as fact checkers? In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 36–41, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jon Roozenbeek, Claudia R Schneider, Sarah Dryhurst, John Kerr, Alexandra LJ Freeman, Gabriel Recchia, Anne Marthe Van Der Bles, and Sander Van Der Linden. 2020. Susceptibility to misinformation about covid-19 around the world. *Royal Society open science*, 7(10):201199.

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.

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What evidence do language models find convincing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7468–7484, Bangkok, Thailand. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Yimeng Lu, Lanyu Shang, Yang Zhang, and Dong Wang. 2024a. Evidence-driven retrieval augmented response generation for online misinformation. 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 5628–5643, Mexico City, Mexico. Association for Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Lanyu Shang, Yifan Liu, Yang Zhang, and Dong Wang. 2024b. Retrieval augmented fact verification by synthesizing contrastive arguments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10331–10343, Bangkok, Thailand. Association for Computational Linguistics.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Yangming Zhou, Qichao Ying, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. 2022. Multimodal fake news detection via clip-guided learning.

A Prompts

```
You are a professional fact checker. To fact-check the
following claim, generate two questions that should be
answered.
You should decompose this claim to two questions to
check its factuality. Don't generate the irrelevant
explanations.
Claim: {query}
Follow the specific JSON format:
{{
        "questions":["question1","question2"]
}}
```

Figure 2: The prompt for decomposing a claim into two questions.

```
You are given a question and relevant documents to the question. Generate
the correct answer to the question and the concise and pointed evidence
based on the documents.
Don't generate any unrelated explanations.
Question: {question} Relevant documents: {documents}
```

Figure 3: The prompt for generating answers to the questions based on top k documents retrieved the from external source.

```
You are a professional fact checker. You are provided with question-answer pairs regarding the
following claim: {query}
Question-Answer pairs:
Question1: {question1}
Answer1: {answer1}
Question2: {question2}
Answer2: {answer2}
Based on strictly the claim and the question-answers provided, you have to provide the rating of
the following claim. You must choose one of the following classes to rate the claim.
Rating: The rating for claim should be one of "supports" if and only if the Question Answer
pairs specifically support the claim, "refutes" if and only if the Question Answer pairs
specifically refutes the claim or "not enough info" if there is not enough information
to answer the claim.
Generated only one rating without any explanations.Follow the JSON format like the example
below:
  {{
      "prediction": "refutes"
  }}
```

Figure 4: The prompt for verifying a claim based on question-answer pairs to gpt-3.5-turbo.

```
instruction=
You are a professional fact checker. You are provided with question-answer pairs
regarding the following claim. Based on the claim and the question-answers provided, you
have to provide the rating of the following claim. You must choose one of the following
classes to rate the claim.
Rating: The rating for claim should be one of "supports" if and only if the Question
Answer pairs specifically support the claim, "refutes" if and only if the Question
Answer pairs specifically refutes the claim or "not enough info" if there is not enough
information to answer the claim appropriately.
prompt=
       Claim : {query}
       Question-Answer pairs:
       Question1: {question1}
       Answer1: {answer1}
       Question2: {question2}
       Answer2: {answer2}
   Does the Question-Answer pairs support,refute, or provide not enough information?
   Generated only one rating without any explanations. Follow the JSON format like the
example below:
  {{
      "prediction": "refutes"
  }}
```

Figure 5: The prompt for verifying a claim based on question-answer pairs to LLaMa.