Retrieving Semantics for Fact-Checking: A Comparative Approach using CQ (Claim to Question) & AQ (Answer to Question)

Nicolò Urbani , Sandip Modha, Gabriella Pasi

Università degli Studi di Milano Bicocca

n.urbani@campus.unimib.it , sandip.modha@unimib.it, gabriella.pasi@unimib.it

Abstract

Fact-checking using evidences is the preferred way to tackle the issue of misinformation in the society. The democratization of information through social media has accelerated the spread of information, allowing misinformation to reach and influence a vast audience. The significant impact of these falsehoods on society and public opinion underscores the need for automated approaches to identify and combat this phenomenon. This paper describes the participation of team IKR3-UNIMIB in AVeriTeC (Automated Verification of Textual Claims) 2024 shared task. We proposed a methods to retrieve evidence in the question and answer format and predict the veracity of a claim. As part of the AVeriTeC shared task, our method combines similarity-based ColBERT re-ranker with traditional keyword search using BM25. Additionally, a recent promising approach, Chain of RAG (CoRAG) is introduced to generate question and answer pairs (QAs) to evaluate performance on this specific dataset. We explore whether generating questions from claims or answers produces more effective QA pairs for veracity prediction. Additionally, we try to generate questions from the claim rather than from evidence (opposite the AVeriTeC dataset paper) to generate effective QA pairs for veracity prediction. Our method achieved an AVeriTeC Score of 0.18 (more than baseline) on the test dataset, demonstrating its potential in automated fact-checking.

1 Introduction

The volume of potentially misleading and false claims has surged with the increasing usage of the web and social media. A study from 2018 found that false information spreads six times faster than the truth on platforms like Twitter (Vosoughi et al., 2018). Considering the democratization of information through social media and the need to limit the spread of misinformation, different approaches and dataset have been explored to automate the fact-checking process (Guo et al., 2022). Over the years, various datasets have been released to facilitate different solutions by the research community. In this context, this paper presents the various approaches for the AVeriTeC (Automated Verification of Textual Claims) Shared Task[1]. The AVeriTeC (Schlichtkrull et al., 2023) dataset contains 5,782 real-world claims (3,068 for training, 500 for development, and 2,214 for testing) sourced from 50 different fact-checking organizations. Each claim is accompanied by annotated question-answer pairs, backed by online evidence, along with textual justifications that explain how the evidence is integrated to produce a verdict. The claims in AVeriTeC are classified into four labels: "Supported", "Refuted", "Not Enough Evidence", and "Conflicting Evidence/Cherry-picking.". The evaluation is performed using the AVeriTeC score, that considers the correct retrieval of evidence and credits veracity predictions only when the correct evidence has been found. The primary goal of the proposed system is to enhance the pipeline by offering pertinent evidence. In this paper, we propose different approaches that focus on the retrieval component through a re-ranking process of the retrieved sentences, aiming to achieve increased precision. Furthermore, unlike the baseline where the question is generated starting from the sentence, we explore whether generating the question from the claim can improve performance. This idea is combined with the methodology proposed in (Khaliq et al., 2024), which utilizes the Chain of RAG (CoRAG), to determine the appropriateness of this technique in this particular case.

Automatic fact-checking has become a fundamental challenge for research. Our paper focuses on analyzing and proposing advanced solutions for fake news detection by enhancing the pipeline with hybrid-search techniques and the use of Chain of RAG (CoRAG).

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

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 37–45

2 Related Work

The early fact checking datasets, such as Fever (Thorne et al., 2018), VitaminC (Schuster et al., 2021), and FEVEROUS (Aly et al., 2021) are assembled from Wikipedia by corrupting the Wikipedia page statements. Gradually, the subsequent datasets like Liar-Plus (Alhindi et al., 2018),PolitiHop (Ostrowski et al., 2021), MultiFC (Augenstein et al., 2019) are developed from the fact checking website. However, these datasets did not address the issues such as context dependence, evidence insufficiency, and temporal leaks. The AVeriTeC dataset (Schlichtkrull et al., 2023) proposed in the shared task has addressed all the above issues during the data assembly.

The baseline of AVeriTeC dataset (Schlichtkrull et al., 2023) consists of several components: (i) **Search**: Using Google API. (ii) **Evidence Retrieval**: Initially, BM25 scores are computed against the claim, and sentences outside the top 100 are discarded. (iii) **Generation of Questions Given Retrieved Evidence**: For each of the retrieved sentences, a question is generated that is answerable by that sentence, using BLOOM with 10 in-context examples in Question and Answer format from the trainset. (iv) **Re-ranking of Retrieved Evidence**: The retrieved evidence in question-answer pair format is re-ranked to identify the top three most relevant pairs for the claim, using a fine-tuned BERT-large model. (v) **Veracity Prediction**: The veracity of the claim is predicted using a fine-tuned BERT-large model with 340 million parameters. (v) **Generation of Justifications**: Justifications are generated using a fine-tuned BART-large model with 406 million parameters.

Different approaches have been used in recent years in the field of fact-checking. The question-answer decomposition is considered a promising strategy, as cited in (Schlichtkrull et al., 2023). A recent work, RAGAR (Khaliq et al., 2024), applies this strategy specifically to political fact-checking, proposing two novel methodologies: **Chain of RAG** (CoRAG) and **Tree of RAG** (ToRAG). This approach appears to improve veracity prediction and the generation of justifications compared to traditional fact-checking. CoRAG employs a sequential question strategy, generating questions and related answers as needed for predicting veracity, using follow-up checks. As evidenced in (Khaliq et al., 2024), these methods have been evaluated as effective and show potential for future research in combating misinformation, suggesting evaluation in other misinformation domains.

Wang et al. (2024) proposed a framework in which evidences are grouped into two groups with opposite polarity with respect to the claim. They used LLM to generate the justification and the inference module for the veracity prediction

To improve the evidence retrieval process, **ColBERT** (Khattab and Zaharia, 2020) is considered due to its effectiveness and efficiency. *ColBERT* is a novel ranking model that adapts deep language models (specifically, BERT) for efficient retrieval with a fine-grained similarity in respect to the provided query.

All the cited works are integrated to develop a solution that achieves high performance in retrieving relevant evidence for accurate veracity prediction.

3 System Description

The **proposed architecture** for the automated fact-checking process can be divided into five parts:

1. Evidence Retrieval
2. Question Generation
3. Evidence Selection
4. Veracity Prediction
5. Justification Generation

Our work primarily focuses on the first three phases of this pipeline. In this section, we provide a detailed description of each component of the proposed system. Two different approaches are used for claim decomposition: generating questions from the answer (AQ) and generating questions from the claim (CQ) directly.

3.1 Answer to Question(AQ)

In the first approach, questions are generated directly from the retrieved evidences, as done in the baseline. A flowchart of the proposed architecture is shown in Figure 1 .

3.1.1 Evidence Retrieval

This is the first phase of the pipeline where evidences are retrieved from the web to verify the claim. We need to find out the sentences from the retrieved evidences that can support or refute the claim. Different strategies can be adopted; in the baseline, a keyword search is applied through BM25. In this work, we introduce a semantic search which is more capable of retrieving information using embedded representations.

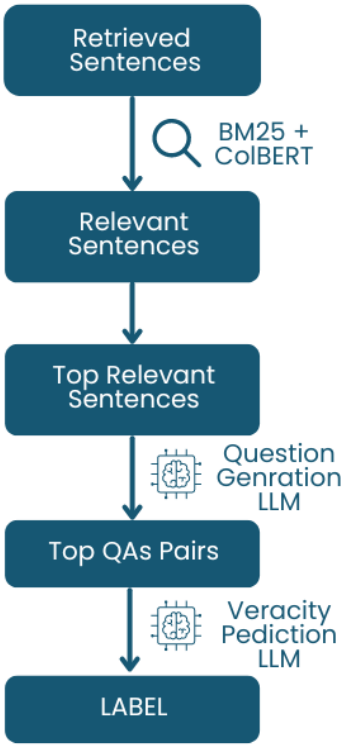

Figure 1: AQ - System Architecture

The representation of the sentences as encoded embedding, it is capable of detecting semantic relations between user queries and sentences. For example, the semantic relation between "cat chases mouse" and "kitten hunts mouse" is stronger than between "cat chases mouse" and "I like to eat ham." For complex semantic text searches, the representations as embedding offers several advantages: capturing semantic similarity and efficiently handling spelling errors and vague descriptions. However, the embedding representation falls short in certain scenarios that require precise matching, such as product names, personal names, product codes, and matching words with low-frequency in the vocabulary. These words often hold significant meaning, such as specific names of people or objects or acronyms.

As both neural and keyword based and searches have their strengths in retrieval, in the proposed method, their combination is achieved by employing ColBERT (Khattab and Zaharia, 2020) for neural search, in addition to BM25 for keyword-based search. In particular, BM25 is used initially to retrieve the most significant sentences through keyword-based matching in relation to the claim to be verified. Then, ColBERT is applied to the retrieved sentences to extract those with closer semantic meaning to the claim.

ColBERT leverages a late interaction architecture that independently encodes both the query and the document using *BERT*. It then applies an efficient yet powerful interaction step to model their fine-grained similarity. This approach enables *ColBERT* to harness the expressiveness of deep language models while significantly accelerating query processing. Remarkably, ColBERT has been demonstrated to be over 170 times faster than traditional BERT-based models, requiring 14,000 times fewer FLOPs per query, with only a minimal reduction in quality. It also outperforms all non-BERT baselines. In our work, ColBERT's ability to retrieve relevant documents quickly is expected to be highly effective, particularly given the large number of claims that need to be processed. This model's capacity for rapid indexing and fast query responses is crucial for managing the extensive data involved in this fact-checking task.

This preserves the ease of use and the effectiveness of keyword-based matching while enabling to extract semantic meaning at a higher level of detail.

3.1.2 Question Generation

The retrieved sentences in the last phase are used to generate evidence in a Question & Answer (QA) format. These sentences become the answers to questions generated by a Large Language Model (LLM). This is achieved using few-shot examples provided to the LLM to ensure that the model generates accurate and relevant questions. Some prompt techniques are used to make the model more effective in generating concise and relevant questions, such as the Chain-of-Thought (CoT) technique that harnesses the power of LLMs to provide logical reasoning steps, deconstruct complex tasks into a sequence of intermediate reasoning steps (Wei et al., 2022). This involves describing all the necessary steps to generate the question starting from the answer, as providing the LLM with a roadmap to follow instead of just the destination.

Two LLMs are considered: BLOOM (Workshop et al., 2022), which is provided in the baseline, and GPT-3.5 (Brown et al., 2020), for a comparison with the baseline.

3.1.3 Evidence Selection

The next step is to re-rank these question-answer pairs to find those most relevant to the claim, by using the fine-tuned BERT-large model (Devlin

et al., 2019), provided in the baseline. This helps in matching the QA pairs that are most similar to the specific claim.

3.1.4 Veracity Prediction

The evidences in QAs format are passed through a BERT-large model (Devlin et al., 2019), with a text classification head, trained on the AVeriTeC dataset, to produce veracity labels. The baseline model is used in this phase, and this step is applied in all the experiments reported below in Section 4.

3.2 Claim to Question(CQ)

Generating the question from the answer (AQ) appears to be an inverse process when we typically ask ourselves questions while reading a claim. For example, when reading the claim "Meatpacking workers have suffered more COVID-19 cases than healthcare workers," certain questions naturally arise. The first question might be, "Were meatpacking workers affected by COVID-19?" followed by, "What is the infection rate in the two sectors?" This process mirrors human reasoning, where a claim prompts us to ask questions. The basic concept is to use the claim to generate questions, and then to extract the most likely response from a knowledge base. We define this approach as Claim to Question Answer (CQ). The goal is to determine which approach performs better: **CQ** (Claim to Question Answer) or **AQ** (Answer to Question). As shown in the flowchart of the proposed system in Figure 2, an LLM generates a question starting from the claim, and the answer is then retrieved from a pre-built ColBERT index. More details are provided in Section 4.3. Furthermore, the strategy recommended for political fact-checking in (Khaliq et al., 2024) is adapted to the current context of general fact-checking by combining a **Chain of RAG** (CoRAG). This approach uses sequential follow-up questions augmented from the RAG response to retrieve further evidence, as described in section 2. This strategy appears to be promising for fact-checking.

4 Experiments

The rationale behind the suggested solution is to apply a Hybrid Search approach combining BM25, as carried out in the baseline, and ColBERT (Khattab and Zaharia, 2020). This preserves the ease of use and effectiveness of keyword-based matching while enabling to extract semantic meaning at a higher level of detail.

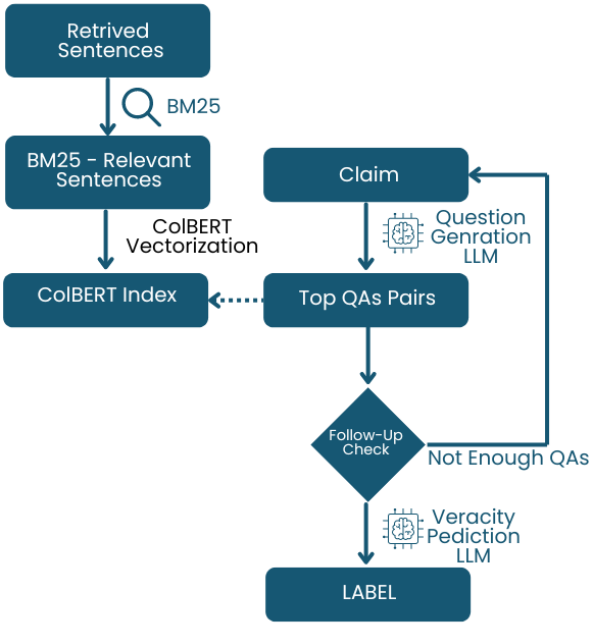

Figure 2: CQ - System Architecture

The next sections introduce the various configurations that are set up to perform the comparative evaluations.

4.1 Run 1 - Basic Hybrid Search

The first run aims to use hybrid Search to improve the baseline. The following techniques have been used to carry out the experiment:

- **Evidence Retrieval** BM25 is used to retrieve the 200 most relevant sentences with respect to the provided claim. Then, by using ColBERT as a re-ranker, the top 10 semantically relevant sentences are selected from the previous top 200 relevant sentences. By using ColBERT to perform a semantic search, the model has a higher likelihood of retrieving relevant documents.

- **Question Generation** The baseline strategy for question generation is applied with BLOOM offering ten comparable samples as a few-shot example, in this case 10 QAs pairs.

- **Question and Answer Re-ranking** In this step, the baseline re-ranker is used to obtain results from 10QAs to 3 QAs.

4.2 Run 2 - GPT as Question Generator

The previous experiment is improved by using a different model for question generation:

- **Evidence Retrieval** In this run, we selected top 3 sentences from the ColBERT reranker's output which substantially help to reduce the computational effort for generating questions in the next phase.

- **Question Generation** Starting with the top three sentences gathered, the pipeline uses GPT-3.5 to generate a question for each sentence, by leveraging a 10-shot example from the prompt. This LLM is employed to compare the GPT model with the baseline's truthfulness prediction. The output consists of three relevant question-answer pairs, which are then used to predict the veracity of the claims.

We have used veracity prediction module of the baseline system for the claim verdict,

4.3 Run 3 - Generating Questions from Claims

In the previously conducted experiments, questions were formed based on the answers retrieved from the sentences; we defined this approach as Answer to Question (AQ). Another approach, Claim to Question Answer (CQ) as defined in section 3.2 is introduced. So the third experiment was conducted following the architecture shown in Figure 2, which is constituted by the following components:

- **Build ColBERT Index** An index is built for each claim by using the previously described hybrid search method; a ColBERT index of the top 200 relevant sentences obtained with BM25, is built by using the claim as query. In other words, for each claim, the top 200 sentences are included in the index. This index will be used to "answer" the question by using the generated question as a query.

- **Generation of QAs from Claim** As previously mentioned, GPT-3.5 is used to generate the question. Starting with a claim and a specific prompt, the model produces a concise question. This question is then used as a query to the ColBERT Index, which retrieves the answer and produces question-and-answer (QAs) pairs. In particular, the Chain of RAG (CoRAG) process is applied as follows:

  – **First Question Generation** In this case, a special prompt, as shown in Appendix A.1, is used since there are no previously generated QAs pairs.

  – **Question Generation** Taking into account the previously generated question and answer, further questions are created using a specific prompt that considers the prior question. This process is detailed in Appendix A.2.

  – **Follow Up Check** A follow-up check is made to the LLM to inquire if the pairs of QAs are sufficient to address the claim veracity; in the event of a negative response, the question generating procedure is repeated, with a maximum of five question-answer pairs permitted. The question-generation process is terminated in the event of a positive response. The propt used for this task is shown in A.3.

  At the end, the procedure yields a sufficient number of QA pairs that can provide useful information for the purpose of veracity prediction.

5 Results

All the experiments introduced and discussed in the previous section are evaluated based on the *AVeriTeC Metrics*, available for both test and development sets. *AVeriTeC Score* measures how closely the generated question and answer pairs match the gold standard, using *Hungarian METEOR* (Banerjee and Lavie, 2005).

In run 1, the performance is better than the baseline, suggesting that the use of semantic search improves the pipeline's effectiveness, as shown in Table 2. In the following runs (run 2 and run 3), questions are generated using GPT-3.5, while in the first experiment, BLOOM is used for question generation. As it may be observed in Table 2, the performance achieved by run 1 is lower compared to those of the other experiments, on both the development and test sets. This suggests that the use of GPT for question generation proves more effective, as shown in Run2-AQ (Answer to Question) and Run3-CQ (Claim to Question). This is reflected by the Q score, a metric that considers the model's ability to generate questions closer to the expected ones, which shows a notable improvement. The results indicate that performance depends significantly on the set of considered claims, warranting further investigation.

In terms of the test set, the performance on the *AVeriTeC score* is higher for Run2-AQ (Answer to Question), indicating that generating questions starting from the answers is more effective, yielding comparable Q and Q+A scores.

In the development set, the F_1 score for each label is available, allowing us to see that the performance is comparable across experiments. In all three runs, the Refuted class has an F_1 score around 0.70, indicating that most of the refuted claims are classified correctly, as shown in Table 1.

Runs	**S**	**R**	**C**	**NEI**	**Macro** F_1
Run1-HS	.43	.71	.09	.00	.30
Run2-AQ	.46	.69	.14	.00	.32
Run3-CQ	.40	.71	.10	.07	.32
Baseline	.41	.69	.10	.16	.23

Table 1: F_1 Score - Development Set Performance (S: Supported, R: Refuted, C: Conflicting Evidence, NEI: Not Enough Information/Evidence)

Considering all the previously discussed factors, we conclude that the best run is Run2-AQ on the test set. This result is variable depending on the evaluated claims and the provided gold question-answer pairs. The obtained *AVeriTeC Score* of 0.18 improves upon the baseline on the test set, increasing from 0.11 to 0.18.

6 Analysis & Discussion

Our improvements to the baseline can be summarized as follows: we enhanced sentence retrieval by focusing on both keyword-based matching and semantic similarity to the claim. This approach allows to capture both keyword-based relevance and semantic alignment, thus retrieving sentences that comprehensively cover various aspects of a claim. This "hybrid search" methodology significantly improves upon the baseline.

Additionally, we explored and compared two approaches for question generation: generating questions from answers versus generating them from claims. Both methods proved effective, but further investigation is needed to determine which is superior. The results indicate that the performance depends significantly on the set of considered claims. The claim-to-question strategy perform better on the development set but unexpectedly underperforms to answer-to-questions on the test dataset. In future, we will try to investigate this inconsistency. A manual check on some claims confirms this: in some cases, the generated questions are more effective in Run2-AQ, where questions are generated starting from the answer, while in other cases, generating the question from the claim, as done in Run3-QA, is more effective. A significant example is the second claim of the test set is the following: "Meatpacking workers have suffered more COVID-19 cases than healthcare workers." In this context, particularly in the E3-CQA experiment, a very useful question to verify the claim is generated: "What is the rate of COVID-19 infection among meatpacking workers in relation to the size of their total workforce compared to the same rate among healthcare workers?". In all other run, for the same claim, questions are generated that consider only healthcare workers or meatpacking workers without comparing the two worker categories together. In other cases, the AQ approach—generating questions from answers—is more effective; it depends on the provided claim.

Generating questions from claims using the Chain of RAG (CoRAG) approach appears to be more practical for real-world applications. Starting from the provided claim, only the questions necessary to verify the claim's veracity are generated, and a corresponding answer is retrieved. In a real-world scenario with a pre-built index, the generated questions are used as queries to retrieve the most suitable answers. This approach reduces the effort required to generate question-answer pairs, thereby shortening the time needed to label claims as 'Supported' or 'Refuted.'

7 Future Work

The AVeriTec dataset contains meta information claim types. We believe that we cannot have the same kind of fact-checking strategy for different types of claims, such as quote verification and comparing numerical quantity. Further investigation is needed to determine whether generating questions from answers or from claims is the more effective approach for the claim type. Additionally, expanding the ColBERT component to include more sentences for vectorization could improve the search's ability to find semantically similar sentences. This is discussed in detail in the limitations section. 8.

	Dev			Test		
	Q	**Q+A**	**Score**	**Q**	**Q+A**	**Score**
Run1-HS	0.24	0.21	0.11	0.25	0.22	0.16
Run2-AQ	0.30	0.21	0.15	0.32	0.23	0.18
Run3-CQ	0.35	0.22	0.16	0.33	0.22	0.16
Baseline	0.24	0.19	0.09	0.24	0.20	0.11

Table 2: Development and Test Set Performance on AVeriTeC Metrics.

8 Conclusion

This paper describes various approaches for the fact verification in the AVeriTeC shared task (Schlichtkrull et al., 2023). The method consists of three main parts: the retrieval component, the question generation component, and the veracity prediction component. In the retrieval phase, sentences are initially retrieved using BM25 and then re-ranked based on their semantic similarity to the claim using ColBERT. For question generation, the previously retrieved sentences are used as answers, with related questions generated through GPT-3.5 in the AQ (Answer to Question) approach. In the second approach, CQ (Claim to Question), the search process is similar, but questions are generated starting from the claim itself. This approach utilizes a recent method called Chain of RAG (CoRAG) to generate question-answer pairs more effectively. Both approaches exhibit effectiveness, with the second approach showing particular promise for real-world applications due to its efficiency in generating questions from claims and building only the necessary Q&A pairs for veracity prediction. Further advancements are needed, including extending the neural re-ranking phase to more sentences to provide the model with a larger pool of potential answers to the proposed questions. Our work contributes to the field by demonstrating the viability of using semantic search and Chain of RAG techniques in fact-checking and suggests further avenues for research in the evidence retrieval task.

Limitations

In all experiments, ColBERT was applied to a **limited number of sentences** due to resource constraints and the volume of claims. In the presented approach, 200 sentences that match the claim with BM25 are considered for building the ColBERT index. Expanding this process to include more than 200 sentences could enhance the model's ability to answer a specific question more accurately and to capture more sentences semantically similar to the claim. The limitations of considering 200 sentences only can be illustrated with the following example. The generated question, "What is the rate of COVID-19 infection among meatpacking workers in relation to the size of their total workforce compared to the same rate among healthcare workers?" could be very useful for verifying the claim. However, it receives the following answer: "Health workers who had > 6 family size were nearly 4 times more likely to be infected with COVID-19 infection compared to healthcare workers who had < 3 family." While this answer is semantically close to the question by considering a comparison between two categories, it does not address the comparison with meatpacking workers. This issue may be related to the limited number of sentences used to answer the query. Additionally, the ColBERT index, as mentioned before, is **built based on sentences more similar to the claim** rather than the question. Therefore, it is possible that the question is not closely related to the claim, making it difficult to find an appropriate answer. Considering this aspect of the proposed system, a possible solution is to extend the ColBERT index with more sentences retrieved using BM25. Another potential approach is to use a fully vectorized knowledge base without using BM25 as a "first filter," although this would require substantial computational resources.

Acknowledgements

This research is supported by the Next Generation EU PRIN 2022 project "20227F2ZN3_001 MoT–The Measure of Truth: An Evaluation-Centered Machine-Human Hybrid Framework for Assessing Information Truthfulness CUP

G53D23002800006”

References

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification, FEVER@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 85–90. Association for Computational Linguistics.

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.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4684–4696. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

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.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-hop fact checking of political claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3892–3898. ijcai.org.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 624–643. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359:1146–1151.

Bo Wang, Jing Ma, Hongzhan Lin, Zhiwei Yang, Ruichao Yang, Yuan Tian, and Yi Chang. 2024. Explainable fake news detection with large language model via defense among competing wisdom. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 2452–2463. ACM.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

A Appendix

In this appendix some details about the used prompt are provided.

A.1 Prompt for First Question Generation

The following instructions were provided to ensure the generation of concise and relevant first question for each claim in approach Claim to Question & Answer (CQA).

```
You are a well-informed and expert fact-
    checker given an unverified claim
    that needs to be explored.
Claim: '''{claim}'''
You follow this instruction:

1. You understand the entire statement.
2. You will make sure that the question
    is specific and focuses on one
    aspect of the statement (focus on
    one topic, detailing where, who, and
     what) and is very, very short.
3. You must not ask for sources of data.
     You are only concerned with the
    question.
4. You are not allowed to use the word '
    claim'. Instead, if you want to
    refer to the statement, you should
    point out the exact issues in the
    statement that you are phrasing your
     question around.
5. You must never ask for calculation or
     methodology.
6. Formulate a pointed fact-check
    question about the statement without
     turning the claim into a question.

Return only the question and nothing
    else.
```

A.2 Prompt for Question Generation

The following instructions were provided to ensure the generation of concise and relevant questions for each claim in approach Claim to Question & Answer (CQA).

```
You are a well-informed and expert fact-
    checker. You are given an unverified
     statement and question-answer pairs
     regarding the issue that needs to
    be explored. You follow these steps:
Statement: '''{claim}'''
Question-Answer Pairs: '''{QAs}'''

Your task is to generate a follow-up
    question regarding the issue
    specifically based on the question-
    answer pairs.
Never ask for sources or publishing.
The follow-up question must be
    descriptive, specific to the issue,
    and very short, brief, and concise.
The follow-up question should not be
    seeking to answer a previously asked
     question. It can, however, attempt
    to improve the question.
You are not allowed to use the word '
    claim' or 'statement.' Instead, if
    you want to refer to the issue, you
    should point out the exact issue in
    the statement that you are phrasing
    your question around.
Formulate a pointed fact-check question
    about the statement without turning
    the claim into a question.

Reply only with the follow-up question
    and nothing else.
```

A.3 Prompt for Follow-up Check

The following instructions were provided for the follow-up check using the Claim to Question Answer (CQA) approach. This method is employed to assess whether the provided evidence, in the form of Questions Answers, is sufficient to verify the veracity of the claim.

```
You are a well-informed and expert fact-
    checker given an unverified claim
    and question-answer pairs regarding
    the claim that needs to be verified.
You follow these steps
Claim: '''{claim}'''
Question-Answer Pairs: '''{QAs}'''
Are you satisfied with the questions and
     you have information to verify the
    claim?
If the answer to any of these questions
    is "Yes",
then replay only with "Yes", or else
    answer, "No"
```