ClaimCheck: Automatic Fact-Checking of Textual Claims using Web Evidence

Akshith Reddy Putta*, **Jacob Devasier, Chengkai Li**
University of Texas at Arlington
{akshith.putta, cli}@uta.edu, jacob.devasier@mavs.uta.edu

Abstract

We introduce ClaimCheck, an efficient fact-checking system that verifies textual claims using smaller, open-source large language models. ClaimCheck integrates two fact-checking strategies, claim-matching and novel claim processing. Claim-matching uses related fact-checks from trusted organizations to fact-check a claim. Novel claim processing breaks down fact-checking into manageable subtasks—generating targeted questions, retrieving Web evidence, extracting answers, and synthesizing verdicts. Evaluation on the AVeriTeC benchmark demonstrates 62.6% verdict prediction accuracy, with claim-matching providing a 2.8% improvement. ClaimCheck approaches the performance of state-of-the-art systems while requiring significantly fewer computational resources, demonstrating the effectiveness of using small language models for fact-checking tasks. Furthermore, our code is publicly available to help make automated fact-checking more accessible.

1 Introduction

The rapid proliferation of digital content has transformed how information is disseminated and consumed, but it has also amplified the spread of misinformation. In an era where public discourse is increasingly influenced by online narratives, combating the impact of false claims has become a critical societal challenge. The World Economic Forum's 2024 Global Risks Report [1] underscores this urgency, identifying misinformation as a top threat to democratic stability, rivaling even climate crises and geopolitical conflicts. As the scale of misinformation grows, so does the necessity for robust, scalable solutions capable of fact-checking claims efficiently (Chen and Shu, 2024).

Automated fact-checking systems have emerged as a promising technological response to this challenge (Dmonte et al., 2024; Vykopal et al., 2024). These systems enhance the efficiency of traditional fact-checking processes by leveraging advancements in machine learning, natural language processing, knowledge bases, and databases (Guo et al., 2022). In this work, we focus on leveraging large language models (LLMs) for fact-checking due to their strong language understanding and reasoning capabilities, as well as their ability to process diverse sources of evidence. LLMs facilitate techniques such as retrieval-augmented generation (RAG) and structured reasoning, which enhance fact-checking capabilities (Khaliq et al., 2024; Iqbal et al., 2024). However, building an effective LLM-based fact-checking system entails overcoming numerous challenges, such as contextual ambiguity, temporal sensitivity of evidence, and incomplete or misleading information (Schumacher et al., 2024; Wang et al., 2024b; Rothermel et al., 2024; Guo et al., 2022).

One of the drawbacks of the current state-of-the-art systems is that most use very large LLMs which can be monetarily prohibitive (Schlichtkrull et al., 2024a). Smaller LLMs require much less computational resources and are more accessible to users at the cost of some loss in task-specific performance and reasoning capabilities (Wang et al., 2024a). We focus this work on utilizing smaller LLMs for the automatic fact-checking pipeline, because this can present more accessible system for the common public. Although the limitations of smaller LLMs impede systems using smaller LLMs from achieving the performance of systems using larger LLMs, we aim to demonstrate that a open-source, less computationally intensive system can be a useful fact-checking system.

This paper presents ClaimCheck, an automatic fact-checking system for textual claims. ClaimCheck first uses a matching process to determine if

*The author is a UTA affiliate and attends Coppell High School.

[1] https://www.weforum.org/publications/global-risks-report-2024/

Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing (KnowledgeNLP'25), pages 303–316

a claim has already been fact-checked. For claims not previously fact-checked, i.e., the given claim is novel, the system decomposes the fact-checking task into targeted subtasks: generating specific questions required to fact-check the claim, retrieving real-time evidence via Web search, processing the evidence to extract answers for each question, and synthesizing these answers to predict a verdict. This design simplifies the subtasks for our system, an important consideration for using smaller LLMs as presenting them with focused tasks helps to prevent them from being overwhelmed by complex instructions and data. Furthermore, our choice of using Web search for evidence retrieval avoids requiring the users to store massive knowledge sources locally for evidence retrieval.

AVeriTeC (Schlichtkrull et al., 2023) is a popular real-world claim benchmark dataset consisting of only textual claims. Our experimental evaluation on AVeriTeC indicates that the proposed approach achieves a verdict prediction accuracy of 62.6%. When compared to the current highest accuracy on AVeriTeC—75.2% (Yoon et al., 2024)—ClaimCheck demonstrates that small, open-source LLMs can be leveraged in a more computationally efficient and scalable manner while approaching the performance of state-of-the-art systems.

To summarize, our work makes the following key contributions to automated fact-checking:

- Our system demonstrates that carefully structured pipelines with small, open-source language models can achieve competitive performance while significantly reducing computational costs compared to systems relying on large language models.
- We incorporate claim-matching into a commonly used framework for automatic fact-checking, resulting in +5.1% increased accuracy on AVeriTeC. These results substantiate the effectiveness of integrating claim-matching with novel claim processing.
- We demonstrate the effectiveness of structured decomposition for fact-checking with smaller LLMs with an accuracy of 62.6% on AVeriTeC.
- The codebase of ClaimCheck is publicly available to help make automated fact-checking more accessible.[2]

[2] https://github.com/idirlab/ClaimCheck

2 Background

To fact-check textual claims, the workflow of most LLM-based systems (Russo et al., 2024; Schlichtkrull et al., 2024b; Braun et al., 2024; Rothermel et al., 2024; Yoon et al., 2024; Niu et al., 2024; Iqbal et al., 2024) contains four steps, as follows. 1) *Question generation*: The system generates questions to identify the core aspects of the claim. This step ensures that the fact-checking process is focused and systematic. 2) *Evidence retrieval*: The system retrieves supporting or refuting evidence from trusted knowledge sources, such as Wikipedia. This step is critical for grounding the fact-checking process with verifiable information. 3) *Question answering*: The system processes the retrieved evidence to generate precise answers to the questions generated in Step 1. This step involves analyzing the evidence and extracting relevant information to address the claim. 4) *Verdict prediction*: The system synthesizes the evidence to predict a verdict (e.g., true or false). The final step determines the overall thurthfulness of the claim.

Fact-checking systems such as ClaimBuster (Hassan et al., 2017) delineate fact-checking strategies, including claim-matching and novel claim processing, which are evidence collection and processing methods to provide a verdict on the truthfulness of the claim. Recent studies (Guo et al., 2022; Iqbal et al., 2024; Niu et al., 2024) have refined this process into LLM-specific tasks. These systems aim to support the functions of traditional fact-checking organizations in addressing misinformation by enhancing efficiency.

Successful textual claim fact-checking requires world and common knowledge, along with some reasoning ability (Rothermel et al., 2024). LLMs have shown to be one of the best tools for these tasks (Rothermel et al., 2024; Schlichtkrull et al., 2024a). Advancements in automated fact-checking have been significantly influenced by the integration of large language models (LLMs) and retrieval-augmented generation (RAG) pipelines. For example, Wang et al. (2025) introduced a framework for LLM-based systems that incorporates an internal mechanism to determine the most suitable LLM for verifying a specific claim. RAGAR (Khaliq et al., 2024) improves fact-checking by leveraging multi-modal inputs and iterative reasoning.

Evidence retrieval methods are important for the credibility and accuracy of automatic fact-checking systems, and are one of the most challenging

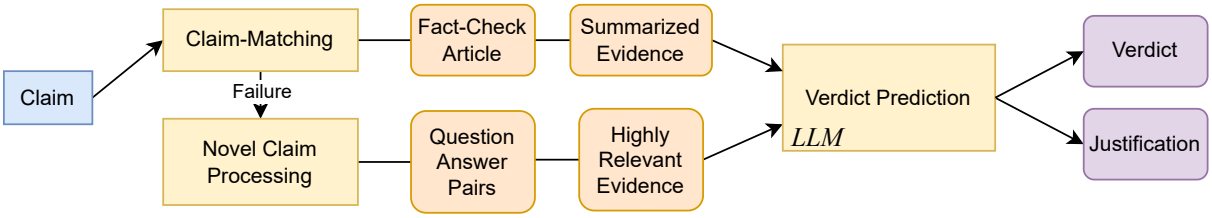

Figure 1: An overview of ClaimCheck. First, the claim is passed into the claim-matching component. If successful, the fact-check article and summarized evidence from the article will be used as evidence. If claim-matching fails, the system proceeds to do novel claim verification, which generates question-answer pairs and summaries of highly relevant evidence to use as evidence. Using previously gathered evidence, an LLM generates a verdict and justification for the claim.

tasks in fact-checking systems (Singal et al., 2024). FactLLaMA (hin Cheung and Lam, 2023) combines pre-trained LLaMA models with external evidence retrieval to validate claims, while Peng et al. (2023) enhances accuracy by integrating external knowledge and providing iterative feedback. Additionally, Singal et al. (2024) tackles misinformation in RAG pipelines by re-ranking retrieved documents based on their credibility scores. Ullrich et al. (2023); Drchal et al. (2023) demonstrate the advantage of using whole documents as evidence to answer questions instead of individual sentences or paragraphs.

For verdict prediction, many fact-checking pipelines use LLM-based verification (Rothermel et al., 2024; Yoon et al., 2024). Finetuning LLMs improves the verdict accuracy (Yoon et al., 2024), and helps avoid inbuilt biases towards certain verdict categories (Rothermel et al., 2024).

3 Methodology

We use two strategies to gather evidence for fact-checking claims: claim-matching and novel claim processing (Figure 1). We first attempt to search for a relevant fact-check for the given claim as we can use them as evidence for the verdict prediction. If a claim has not been previously fact-checked, it is considered novel. For novel claims, we break down the fact-checking process into these key steps: claim reformulation, question generation, query generation, online evidence retrieval, question answering and evidence curation. Using the evidence collected from either claim-matching or novel claim processing, an LLM is used to predict the veracity of the claim along with a justification. For novel claim processing, we only use online search so that our system is applicable to real-world uses.

3.1 Claim-Matching

Fact-Check Article Retrieval Our system first does a Web search using the Google Search API,[3] with the claim being the search query. Then, articles published after the claim was made are excluded to prevent data leakage. To ensure the accuracy and reliability of claim-matching, our system only uses fact-checks from well-established and reputable sources. These sources include global fact-checking initiatives such as Africa Check and AFP, regional fact-checkers such as factcheck.kz and factcheck.ge, and widely recognized fact-checking platforms such as PolitiFact, and Snopes.[4]

Article Summarization Next, given each retrieved fact-check article, the LLM is prompted, with Listing 1, to check if the article is relevant. If the article is relevant, the LLM produces a summary of relevant evidence from the article and how it can clearly lead to a verdict, and ClaimCheck uses the collected evidence in its verdict prediction step. If the article is not relevant, the next article from the Google search results is sequentially presented to the LLM. If no useful articles are detected within the search results, the system proceeds to the novel claim processing (Figure 2).

3.2 Novel Claim Processing

Claim Reformulation Our claim reformulation step ensures that the claim is ready for question generation by augmenting the claim with the claim's date of origin, the author of the claim (claimant), and the URL of the claim's origin, which are all provided in the AVeriTeC dataset. The LLM is

[3] https://programmablesearchengine.google.com/

[4] The full list of the sources is africacheck.org, factcheck.kz, altnews.in, boomlive.in, vishvasnews.com, factcheck.ge, poynter.org, factcheck.afp.com, apnews.com, reuters.com, checkyourfact.com, hoax-slayer.net, leadstories.com, fullfact.org, truthorfiction.com, politifact.com, and snopes.com.

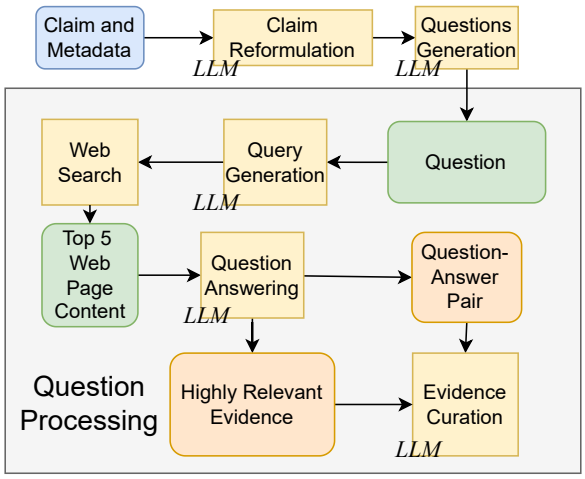

Figure 2: Overview of novel claim processing. The claim and metadata from AVeriTeC are used to reformulate the claim. Next, questions are generated to fact-check the claim. A Web search query is generated for each question. For the question answering step, the content of the top 5 search results are given sequentially to an LLM to answer the question. If an answer has been found using one of the results' content, a question-answer pair will be created. If a result's content is highly relevant evidence, but does not answer the question, it is also saved. The QA pairs and highly relevant evidence are then checked for their relevance.

prompted to reformulate the claim based on the supplied metadata rather than its knowledge from training data, as doing so may introduce hallucinations, particularly when using small LLMs. The prompt is provided in Listing 2.

Question Generation Next, ClaimCheck generates questions that are essential to fact-checking the reformulated claim. To generate these questions we prompt an LLM using Listing 3 with three static claims and their corresponding questions from the AVeriTeC dataset. The generated questions will be specific to each claim, avoiding generalized questions such as "when was this claim made?", which could lead to retrieving unusable evidence. We do not specify a number of questions, to avoid unnecessarily generating similar questions, which could lead to retrieving the same evidence repeatedly, or not generating enough questions. Figure 3 displays a few example questions generated by the LLM for a claim. Some previous fact-checking frameworks (Rothermel et al., 2024) have set a requirement for the number of questions, which could result in the same evidence being repeatedly retrieved multiple times, unnecessarily using computational resources. This is usually due to similar questions retrieving the same evidence.

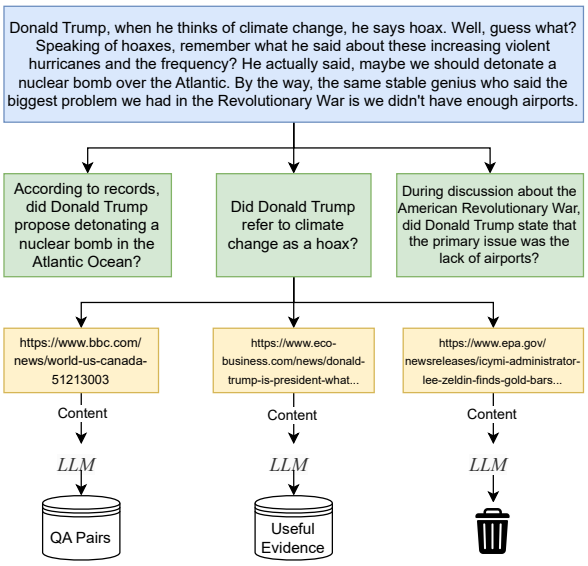

Figure 3: Example of question generation and question answering processes.

Query Generation For each generated question, ClaimCheck uses an LLM to generate a Web search query, using Listing 4. The query is a reformulation of the question, incorporating the claim and its metadata. This process is essential, as directly using the questions as the Web search queries often results in excessively long queries that hinder effective evidence retrieval.

Online Evidence Retrieval Evidence retrieval employs a Google search using the Serper API [5] to locate relevant information, where the generated queries are used. ClaimCheck takes into account temporal constraints to ensure evidence validity, excluding evidence posted or updated after the claim date. We use the top 5 webpages as evidence.

Question Answering The question-answering (QA) phase is illustrated in Figure 3. ClaimCheck analyzes the evidence retrieved to answer the questions generated in the previous step, using Listing 5. The LLM is prompted with the content of the website, extracted using Trafilatura (Barbaresi, 2021) in the online evidence retrieval step, the relevant question, and the claim itself. The LLM has three choices:

1. It can answer the question using the evidence provided, if the evidence completely answers the question, and the system moves on to the next question.
2. It can decide that the evidence does not answer the question but is highly relevant for

[5] https://serper.dev/

fact-checking the claim, in which case the website content is saved, and the next piece of evidence is presented.
3. It can also decide the evidence is not helpful for answering the question nor is highly relevant for fact-checking the claim, in which case the evidence is rejected, and the next piece of evidence is presented.

If all pieces of evidence are rejected, the question is not answerable, and this outcome is passed onto verdict prediction as evidence.

Evidence Curation All QA pairs and highly relevant evidence pieces are checked for relevance to the claim. The issue of irrelevant evidence might arise due to the limitations of smaller LLMs, which may generate summaries of the online evidence even when there is no connection to the claim. The LLM iterates through all QA pairs and retains only those useful for fact-checking, as instructed in the prompt (Listing 6). Evidence is considered relevant if its content directly pertains to the claim. For such evidence, the LLM generates a summary; otherwise, it is discarded. The relevant QA pairs and summarized highly relevant evidence pieces are sent to the verdict prediction step of ClaimCheck.

3.3 Verdict Prediction

Once all the evidence is gathered, ClaimCheck uses the LLM to produce a verdict prediction, assigning the claim to a verdict that could be reasonably assumed using the evidence present. The verdicts that can be predicted are Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking, from Schlichtkrull et al. (2024a). The LLM simultaneously creates a justification to enhance the system's interpretability. The LLM creates textual explanations detailing how the retrieved evidence supports the final verdict. This feature makes the system's reasoning transparent and understandable to users. To avoid out-of-memory errors for the LLMs, we truncate the evidence at the maximum context length. The LLM is instructed to provide the verdict and justification in JSON format, to avoid issues with the LLM not returning the required information (Shorten et al., 2024).

We use Qwen2.5-7B for all tasks other than fact-verification, while a fine-tuned Qwen2.5-7B model is used for verdict prediction, which requires more reasoning and decision-making ability. For fine-tuning, we used a 4 bit-quantized Qwen2.5-7B model and performed the training using LoRA (Hu et al., 2022) with rank $r = 16$. The training set of AVeriTeC was used for fine-tuning. The models' temperature and top p are the default Ollama [6] parameters for the respective models. We only fine-tuned for the verdict prediction step. Listing 7 presents the prompt using AVeriTeC's verdict classes, but ClaimCheck is adaptable and can support alternative verdict categories.

4 Experiments

4.1 Dataset

Schlichtkrull et al. (2024a) introduced a novel automated fact-checking system utilizing the AVeriTeC dataset, a rich resource comprising 4,568 claims drawn from 50 fact-checking organizations. This dataset classifies the claims into the 4 verdicts mentioned in Section 3.3. AVeriTeC includes not only claims but also annotated question-answer pairs, and justifications, making it a valuable benchmark for real-world fact verification tasks. It also includes a knowledge store, which consists of approximately 1000 pieces of evidence per claim. The knowledge store was created by generating multiple queries related to each claim, conducting Web searches for these queries, and saving the top search results.

4.2 Experiment Setup

ClaimCheck was evaluated using the development subset of the AVeriTeC dataset, which consists of 500 claims. Throughout the experiments, particular attention was paid to avoiding common pitfalls in automated fact-checking, particularly temporal leakage. We excluded fact-check articles which were published after the claim date, and only searched for evidence posted before the claim date. Our metric is claim alignment accuracy, which measures the proportion of predicted verdicts matched to the gold verdicts in AVeriTeC. It is calculated by dividing the number of correctly predicted verdicts by the total number of claims.

4.3 Results

Table 1 presents the end-to-end accuracies of multiple fact-checking systems. We have run naive predictions on multiple models, to show the baseline fact-checking ability of the LLMs. This is when we prompt LLMs to give a verdict without any evidence, solely relying on the LLM's internal knowledge. Table 2 is an ablation of the verdict

[6] https://ollama.com/

Framework	Accuracy	Evidence Source
ClaimCheck	**0.626**	Online Search
ClaimCheck without claim-matching	0.598	Online Search
Papelo	0.415	Online Search
HerO	**0.752**	Knowledge Base
InFact	0.724	Knowledge Base
Naive GPT-4o	0.532	N/A
Naive GPT-4o-Mini	0.468	N/A
Naive Qwen2.5-7B	0.260	N/A

Table 1: Claim alignment accuracy of different frameworks on the AVeriTeC dataset.

LLM	Accuracy
Fine-tuned Qwen2.5-7B	**0.626**
Phi-4	0.494
GPT-4o	0.396
GPT-4o-mini	0.314
Qwen2.5-7B	0.280

Table 2: Claim alignment accuracy of ClaimCheck framework using different LLMs for verdict prediction.

Evidence Retrieval Component	Proportion
Claims with evidence	0.980
Claims with evidence (after evidence curation)	0.696
Questions answered	0.949
Fact-check articles matched	0.158
Claim-matching only accuracy	0.759

Table 3: Performance measures for evidence retrieval components in ClaimCheck.

prediction step of ClaimCheck, evaluating the performance of larger and smaller LLMs along with a fine-tuned small LLM. Finally, Table 3 presents performance measures for the evidence retrieval systems, highlighting their role in supporting accurate verdict prediction.

Team Papelo (Malon, 2024) achieved the highest accuracy at FEVER-24 of the frameworks using online search with a focus on claim decomposition and iterative searching. Their approach involves an initial search followed by targeted queries to fill information gaps. In contrast, ClaimCheck employs a single-pass system to collect evidence. Another significant difference to ClaimCheck is the computational approach—while Team Papelo's system relies on larger LLMs (GPT-4o) for sophisticated reasoning, ClaimCheck demonstrates the viability of using smaller, open-source LLMs (Qwen2.5-7B) through careful task decomposition and structured verification steps, making it more accessible and computationally efficient. However, Team Papelo only predicts whether a claim is supported or refuted, without predicting other verdicts, in their final system. For the purposes of comparison, we use Team Papelo's accuracy on four classes. Additionally, we present the accuracy of the best-performing systems at FEVER-24, HerO and InFact, according to the claim alignment accuracy, which use all four classes (Yoon et al., 2024; Rothermel et al., 2024).

The experiment results demonstrate the significant impact of fine-tuning on verdict prediction performance across different language models, with the marked improvement of 36.6 percentage points. The fine-tuned Qwen2.5-7B model achieved the highest accuracy at 62.6%, substantially outperforming its non-fine-tuned counterpart which scored only 26.0%. Other models showed varying degrees of performance, with Phi-4 achieving 49.4% accuracy, followed by GPT-4o at 39.6%, and GPT-4o-mini at 31.4%.

4.4 Error Analysis

Analysis of 30 incorrectly predicted samples showed that a common issue was the useful evidence being retrieved and the LLM interpreted it properly, but it gave the wrong verdict. For the claim "Most deaths in the 1918 influenza pandemic originated from bacterial pneumonia caused by face masks and that Dr Anthony Fauci, the US government's top expert in the fight against Covid-19, knew about it.", the LLM responded in the verdict prediction that "The evidence from the fact-check supports the claim that face masks did not directly cause most deaths in the 1918 influenza pandemic.", with other supporting evidence, but due to misinterpreting the claim, it responded with Supported. It had enough information to completely fact-check

the claim, but it gives an incorrect verdict. Another issue is that the LLM sometimes just provides a justification instead of a verdict, even when prompted that it must produce a verdict. These issues are the most common cause of errors, and they are not due to ClaimCheck's system architecture. To address problems with verdict prediction, reasoning models fine-tuned on a large corpus of fact-checks could enhance LLMs' understanding of fact-checking procedures and improve verdict prediction performance.

Another source of error with ClaimCheck is the evidence curation. The evidence curation step is necessary due to the models not being able to judge evidence relevance when doing verdict prediction, but this sometimes results in useful evidence being excluded. Moreover, the lack of support to use image and video evidence hinders ClaimCheck's ability to fact-check some claims, particularly where quote or action verification is required.

When fact-check articles are retrieved by the Web search, the article might contain fact-checks of multiple claims. The LLM might use one of the other fact-checks as evidence instead of the fact-check pertaining to the claim, leading to incorrect evidence being used for verdict prediction. Similarly, another issue is irrelevant evidence making it past the evidence curation stage, which overloads the LLM with information, leading to incorrect verdict prediction. These are issues with the smaller LLMs, due to their limited reasoning capability (Wang et al., 2025).

5 Discussion

5.1 System Architecture Trade-offs

The architectural framework of ClaimCheck represents a significant departure from contemporary state-of-the-art systems such as InFact (Rothermel et al., 2024) and HerO (Yoon et al., 2024). Whereas these established systems rely on pre-collected knowledge bases, ClaimCheck implements a dynamic Web search methodology that facilitates real-time information access and enhanced temporal processing. This approach requires careful consideration of the associated challenges, such as the system's occasional retrieval of extraneous or redundant information, necessitating the evidence curation step, which can impact system performance.

Evidence Retrieval Quality It can be concluded that the evidence retrieval system is functioning effectively, as all claims using novel claim processing have supporting evidence, and 98% of questions are fully answered. However, for 28.4% of claims, all of the retrieved evidence is discarded. This might suggest that the major bottleneck in the ClaimCheck system is the evidence retrieval system, as relevant evidence is not being retrieved.

Claim-Matching Strategy The empirical effectiveness of the claim-matching component, successfully processing 15.8% of claims, as shown in Table 3, demonstrates the value of leveraging existing fact-checks. This methodological enhancement, notably absent from FEVER-24 submissions, yields a noticeable improvement in accuracy (+2.8%) compared to using novel claim processing only for all claims. Fact-check articles could possibly even be useful for checking novel claims, as this could give valuable context about the claims. The success of this approach indicates that future system architectures might benefit from implementing a hybrid methodology that synthesizes both pre-existing fact-checks and real-time evidence acquisition.

Question Generation Allowing the LLM to generate questions without specifying a required number of questions appears to have increased the quality of the questions. However, this has also led to aspects of the claim not being fact-checked correctly, due to the nature of small LLMs. This could possibly be solved by fine-tuning or reinforcement learning using the small LLMs.

Another approach considered was leveraging the misinformation sites provided by AVeriTeC, along with some satire sites, to identify claims originating from these sources. This could offer a straightforward method to flag potential misinformation. However, in the development set, no claims were found to originate from these sources, and only 52.4% of all claims had origin URLs available. This suggests that such an approach may have limited effectiveness due to the low availability of origin URLs and the lack of overlap between claims and known misinformation sources.

5.2 Small Language Models in Fact-Checking

The usage of small LLMs in fact-checking tasks has both significant potential and inherent limitations. While these models offer considerable advantages in terms of computational efficiency and accessibility, they are subpar for complex reasoning tasks. In the error analysis, we have noticed that smaller

Claim	Predicted Verdict	LLM-generated Justification	Error
Even if your mail-in ballot is postmarked prior to the Election day but arrives after the Election day it will be thrown out.	N/A	The article discusses potential risks associated with following Trump's advice during the election, particularly concerning long lines at polling places. Experts note that such behavior could exacerbate existing infrastructure challenges posed by the pandemic, indicating that this scenario is plausible and thus supports the idea of chaos on Election Day.	No verdict generated.
Illegal drug, border crossings, and human smuggling activities have decreased in areas where barriers are deployed in the United States.	Refuted	This claim is refuted. According to the QA Pairs provided it seems that there was no evidence found for this claim.	Incorrect understanding of verdict definitions.
Donald Trump said that $15 an hour is too much for essential workers	Refuted	The claim states that Joe Biden commented on Donald Trumps statement. There was no proof of this in the QA pairs, Therefore it's refuted.	Incorrect claim interpretation.
There is a scam involving Santander customers being sent fake bank cards.	Refuted	Q&A 1 gives no evidence for the claim so it is refuted.	Uses incorrect verdict.

Table 4: Example of LLM-caused errors in verdict prediction.

LLMs sometimes do not follow instructions when the prompt is very large, particularly for question answering and verdict prediction, both being RAG tasks. This can result in incorrect claim interpretation or not recognizing unusable evidence, which necessitated the evidence curation step.

The verdict generation phase was challenging for small language models. The error analysis revealed instances where the LLM produced justifications on the claim without providing a verdict. These findings suggest that while small language models can effectively manage generation tasks within the fact-checking pipeline, their performance is subpar for retrieval tasks such as question answering. This might indicate that larger LLMs, finetuned smaller LLMs, or RAG-specific language models might perform better for this step. Examples of common errors made by LLMs in verdict prediction are given as examples in Table 4.

Notably, the fine-tuned Qwen2.5-7B model's superior performance suggests that fine-tuning can help overcome some of the inherent limitations of smaller language models in complex reasoning tasks like verdict prediction. The major issue for non-fine-tuned models is the tendency for models to select Not Enough Evidence even when there is enough evidence to reach a verdict. In Malon (2024), only the Supported and Refuted classes were the only classes the LLM could predict.

5.3 Real-World Applicability

The system design underlying ClaimCheck is to demonstrate the effectiveness of small LLMs on real-world claims, such as those in AVeriTeC. The use of small, open-source language models could help mitigate the spread of misinformation on social media. The Web evidence retrieval can more easily handle novel claims, compared to a static knowledge base, particularly on claims about recent events. This makes it better suited for fact-checking the rapidly evolving claims found online.

This choice of using small LLMs presents distinct challenges. While offering enhanced efficiency, small language models necessitate more tasks in the fact-checking pipeline, which could lead to more sources of error. The system's LLM-agnostic design allows for improvements as LLM capabilities advance. Nevertheless, the results suggest that accessible fact-checking tools utilizing small language models can provide substantial support for fact-checking tasks, particularly when integrated with claim-matching.

6 Conclusion

ClaimCheck demonstrates the viability of Web evidence retrieval for automatic fact-checking systems using smaller language models. It achieved 0.626 accuracy on the AVeriTeC benchmark dataset. Our approach establishes essential procedures for end-to-end fact-checking systems without relying on resource-intensive larger models. By developing an LLM-size agnostic process, we ensure that ClaimCheck can benefit from future LLM advancements while maintaining independence from specific model designs. We additionally show that claim matching can be a useful evidence retrieval approach to fact-checking non-novel claims.

The use of small, open-source LLMs enhances

reproducibility and accessibility. Our online search mechanism efficiently leverages external search algorithms to retrieve only the most relevant evidence, significantly reducing computational resource demands compared to retrieving and analyzing information from knowledge bases. However, challenges with evidence quality necessitated our multi-question approach and content curation task.

Future work could explore iterative systems rather than single-pass frameworks, incorporate multimedia analysis capabilities for social media claims, and investigate targeted fine-tuning approaches that balance performance improvements with system independence.

Limitations

The current implementation of ClaimCheck exhibits several significant limitations that warrant consideration. The system's inability to process non-textual information substantially restricts its efficacy in addressing social media claims, where misinformation frequently propagates through visual media. The Web-based evidence retrieval system, while providing access to current information, occasionally yields irrelevant or unreliable sources that may compromise verification accuracy. Furthermore, the system's dependence on English-language fact-checking websites introduces limitations in global applicability.

Ethics and Risks

Beyond technical constraints, the system's reliance on fact-checking websites raises substantial ethical considerations. The selection criteria for trusted fact-checking domains may introduce systematic biases in evidence selection. Moreover, the automated nature of the system could potentially lead to excessive reliance on machine-generated verdicts without appropriate human oversight. Future research directions should address these limitations while maintaining system accessibility and efficiency, potentially through the implementation of enhanced source validation mechanisms and support for multiple languages and modalities.

Acknowledgements

We extend our gratitude to the Texas Advanced Computing Center (TACC) for providing computational resources used in this work's experiments.

References

Adrien Barbaresi. 2021. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131, Online. Association for Computational Linguistics.

Tobias Braun, Mark Rothermel, Marcus Rohrbach, and Anna Rohrbach. 2024. DEFAME: Dynamic evidence-based fact-checking with multimodal experts. *arXiv preprint arXiv:2412.10510*.

Canyu Chen and Kai Shu. 2024. Can LLM-generated misinformation be detected? In *The Twelfth International Conference on Learning Representations*.

Alphaeus Eric Dmonte, Roland Oruche, Marcos Zampieri, Prasad Calyam, and Isabelle Augenstein. 2024. Claim Verification in the Age of Large Language Models: A Survey. *ArXiv*, abs/2408.14317.

Jan Drchal, Herbert Ullrich, Tomás Mlynár, and Václav Moravec. 2023. Pipeline and Dataset Generation for Automated Fact-checking in Almost Any Language. *ArXiv*, abs/2312.10171.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. ClaimBuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10:1945–1948.

Tsun hin Cheung and Kin Man Lam. 2023. FactLLaMA: Optimizing instruction-following language models with external knowledge for automated fact-checking. *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 846–853.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Nenkov Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2024. OpenFactCheck: A unified framework for factuality evaluation of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 219–229, Miami, Florida, USA. Association for Computational Linguistics.

Mohammed Abdul Khaliq, Paul Yu-Chun Chang, Mingyang Ma, Bernhard Pflugfelder, and Filip Miletić. 2024. RAGAR, your falsehood radar: RAG-augmented reasoning for political fact-checking using multimodal large language models. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 280–296, Miami, Florida, USA. Association for Computational Linguistics.

Christopher Malon. 2024. Multi-hop evidence pursuit meets the web: Team papelo at FEVER 2024. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 27–36, Miami, Florida, USA. Association for Computational Linguistics.

Cheng Niu, Yang Guan, Yuanhao Wu, Juno Zhu, Juntong Song, Randy Zhong, Kaihua Zhu, Siliang Xu, Shizhe Diao, and Tong Zhang. 2024. VeraCT scan: Retrieval-augmented fake news detection with justifiable reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 266–277, Bangkok, Thailand. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv*.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. InFact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112, Miami, Florida, USA. Association for Computational Linguistics.

Daniel Russo, Stefano Menini, Jacopo Staiano, and Marco Guerini. 2024. Face the facts! Evaluating RAG-based fact-checking pipelines in realistic settings. *Preprint*, arXiv:2412.15189.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024a. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos, editors. 2024b. *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*. Association for Computational Linguistics, Miami, Florida, USA.

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

Dan Schumacher, Fatemeh Haji, Tara Grey, Niharika Bandlamudi, Nupoor Karnik, Gagana Uday Kumar, Jason Cho-Yu Chiang, Paul Rad, Nishant Vishwamitra, and Anthony Rios. 2024. Context matters: An empirical study of the impact of contextual information in temporal question answering systems. *Preprint*, arXiv:2406.19538.

Connor Shorten, Charles Pierse, Thomas Benjamin Smith, Erika Cardenas, Akanksha Sharma, John Trengrove, and Bob van Luijt. 2024. StructuredRAG: JSON response formatting with large language models. *Preprint*, arXiv:2408.11061.

Ronit Singal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 91–98, Miami, Florida, USA. Association for Computational Linguistics.

Herbert Ullrich, Jan Drchal, Martin Rýpar, Hana Vincourová, and Václav Moravec. 2023. CsFEVER and CTKFacts: acquiring Czech data for fact verification. *Language Resources and Evaluation*, 57(4):1571–1605.

Ivan Vykopal, Matúš Pikuliak, Simon Ostermann, and Marián Simko. 2024. Generative large language models in automated fact-checking: A survey. *ArXiv*, abs/2407.02351.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, Qi He, Yao Ma, Ming Huang, and Suhang Wang. 2024a. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with LLMs, and trustworthiness. *ArXiv*, abs/2411.03350.

Ning Wang, Yun Xiao, Xiaopeng Peng, Xiaojun Chang, Xuanhong Wang, and Dingyi Fang. 2024b. ContextDet: Temporal action detection with adaptive context aggregation. *Preprint*, arXiv:2410.15279.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2025. OpenFactCheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11399–11421, Abu Dhabi, UAE. Association for Computational Linguistics.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. 2024. HerO at AVeriTeC: The herd of open large language models for verifying real-world claims. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 130–136, Miami, Florida, USA. Association for Computational Linguistics.

Listing 1: Prompt for Article Summarization

```
Can this fact-checking article provide a complete fact-check for the claim, including a clear verdict and
    justification with relevant evidence?
Take into account the claim date and any other information important for fact-checking the claim.

Possible Verdicts:
- Supported: The knowledge from the fact-check supports or at least strongly implies the claim. Mere
    plausibility is not enough for this decision.
- Refuted: The knowledge from the fact-check clearly refutes the claim. The mere absence or lack of
    supporting evidence is not enough reason for being refuted (argument from ignorance). This includes
    fake news and deliberate misinformation.
- Conflicting Evidence/Cherrypicking: The knowledge from the fact-check contains conflicting evidence from
    multiple reliable sources. Even trying to resolve the conflicting sources through additional
    investigation was not successful.

Claim: {claim}
Article: {article_text}
If the article cannot fulfill this requirement, respond with "No answer found." Otherwise, gather the key
    evidence from the article that can be used for fact checking the claim and summarize them in at most
    one paragraph.
```

Listing 2: Prompt for Claim Reformulation

```
# Instructions
You are presented with a raw claim, with additional metadata like Content date or speaker. **Your task right
     now is to interpret the claim.** That is, identify the speaker's core message and write down the main
    point(s) using your own words. Do not ask any questions and only use the metadata provided to interpret
     the claim. Be concise and write only one paragraph.

# Content
Original Claim: {claim}
Metadata:
    - Speaker: {speaker}
    - Date: {claim_date}
    - Origin URL: {original_claim_url}
    - Reporting Source: {reporting_source}
    - Location ISO Code: {location_ISO_code}

# Interpretation
```

Listing 3: Prompt for Question Generation

```
# Instructions
You are a fact-checker verifying a claim. Your task is to generate clear, specific, and relevant fact-
    checking questions that help assess the accuracy of the claim.

**Guidelines:**
- Focus on the essential details of the claim. The questions should help find direct evidence to confirm or
    refute it.
- Only use metadata (such as date, speaker, or source) when it is necessary for verification (e.g., when
    time-sensitive or quote verification is in question).
- Each question should be concise and directly related to the claim.
- Format each question using backticks like `this`.
- Do not repeat questions already addressed in prior fact-checking records.

**Examples:**
Claim: "New Zealand's new Food Bill bans gardening."
Questions:
1. Does New Zealand's Food Bill ban home gardening?
2. What are the key regulations in the New Zealand Food Bill related to gardening?
3. Has the New Zealand government enforced any gardening restrictions under this bill?

Claim: "Video of a man blowing vape smoke through various face masks shows that they do not help prevent the
     spread of coronavirus."
Questions:
1. How does coronavirus spread?
2. Do scientific studies show that face masks reduce the spread of coronavirus?
3. Does the ability of vape smoke to pass through a mask indicate ineffectiveness against viruses?

Claim: "The Nigerian government is donating $600 million to Democratic presidential nominee Joe Biden's
    campaign."
Questions:
1. Is there evidence that the Nigerian government donated $600 million to Joe Biden's campaign?
2. Are foreign governments legally allowed to donate to U.S. presidential campaigns?
3. Has the Biden campaign reported any donations from Nigeria?

# Claim to Verify
Claim: {claim}
Metadata: {metadata}

## Questions:
```

Listing 4: Prompt for Query Generation

```
# Instructions
You are a fact-checker optimizing a question for web search to retrieve relevant evidence.

**Guidelines:**
- Ensure the query makes sense in the context of the question.
- Add claim-specific context only if absolutely necessary to improve relevance.
- Keep the query concise and structured for effective search results.
- Format the final query using backticks like `this` (without extra formatting or explanation).

## Question
{question}

## Claim
{claim}

## Search Query:
```

Listing 5: Prompt for Question Answering

```
Instructions
You are a fact-checker. Your overall motivation is to verify a given Claim. In order to find evidence that
    helps the fact-checking work, you just ran a web search which yielded a Search Result. Your task right
    now is to answer the Question given below. Adhere to the following rules:

The length of your Answer should be between one sentence and one paragraph.
If applicable and useful, you may directly cite relevant excerpts from the source. In that case, put the
    citation into quotation marks.
If the search result does not contain sufficient information to answer the Question or is unrelated to the
    question completely, respond simply with Answer Not Found.
If the evidence does not answer the question, but can otherwise be highly useful for the fact-check, you
    must respond with "The evidence is useful, but does not answer the question." This is a very rare case.

Claim: {claim}

Question
{question}

Search Result
Summary: {snippet}

Evidence:
{evidence_text}

Your Answer
```

Listing 6: Prompt for Evidence Curation

```
Instructions
You are a fact-checker. Your overall motivation is to verify a given Claim. In order to find evidence that
    helps the fact-checking work, you just ran a web search which yielded a Search Result. Your task right
    now is to determine if the Answer is useful to fact-checking the Claim. Follow the following rules:
An answer is useful even when it doesn't directly answer the question, if it provides highly relevant
    information for fact-checking. It just has to be somewhat related to the Claim.
If the Answer is useful to fact-checking the Claim, respond only with "Yes".
If the Answer is not useful to fact-checking the Claim, respond only with "No".

Claim: {claim}

Question and Answer: {answer}
```

Listing 7: Prompt for Verdict Prediction

```
# Fact-Checking Analysis Task

## Objective
Analyze the provided evidence and QA pairs to determine the veracity of the claim using the structured
    methodology below.
Must output the data in the structured JSON format, not just as text. The verdict must be one of the
    following options: Supported, Refuted, Conflicting Evidence/Cherrypicking, Not Enough Evidence.
---

## Verification Protocol

1. **Evidence Synthesis**
   - Identify factual anchors in both evidence and QA responses
   - Note contradictions, corroborations, and evidence quality

2. **Verdict Determination**
   Select ONE of the below verdicts using these strict criteria:

   **Supported**
   - Evidence conclusively proves claim true
   - Multiple credible sources align without contradiction

   **Refuted**
   - Evidence disproves central claim elements
   - Includes fabricated content/deceptive practices
   - Lack of any credible sources supporting the claim

   **Conflicting Evidence/Cherrypicking**
   - Reputable sources directly contradict each other
   - No resolvable consensus after analysis

   **Not Enough Evidence**
   - No relevant evidence found after exhaustive search
   - Claim too vague for substantive evaluation
   *(Last-resort option only)*

   Do not select any other verdicts.

---

## Input Data
**Claim to Evaluate**
{claim}

**Relevant Evidence**
{relevant_evidence}

**QA Pair Analysis**
{qa_text}

---

## Output Requirements

Must output the data in the following JSON format, no exceptions.:

**JSON Structure**
```json
{{
    "classification": "One of the above verdict options",
    "justification": "Cohesive analysis paragraph of reasoning for the selected verdict"
}}
```
Example Output:

```json
{{
    "classification": "Refuted",
    "justification": "The evidence and answers show that the claim was published on a fake news site, so the
        claim is refuted."
}}
```
```