PASS-FC: Progressive and Adaptive Search Scheme for Fact Checking of Comprehensive Claims

Anonymous ACL submission

Abstract

Automated fact-checking faces challenges in handling complex real-world claims. We present PASS-FC, a novel framework that addresses these issues through claim augmentation, adaptive question generation, and iterative verification. PASS-FC enhances atomic claims with temporal and entity context, employs advanced search techniques, and utilizes a reflection mechanism. We evaluate PASS-FC on six diverse datasets, demonstrating superior performance across general knowledge, scientific, real-world, and multilingual fact-checking tasks. Our framework often surpasses stronger baseline models. Hyperparameter analysis reveals optimal settings for evidence quantity and reflection label triggers, while ablation studies highlight the importance of claim augmentation and language-specific adaptations. PASS-FC's performance underscores its effectiveness in improving fact-checking accuracy and adaptability across various domains. We will open-source our code and experimental results to facilitate further research in this area.

1 Introduction

The proliferation of online information and the advent of large language models (LLMs) have significantly increased the volume and complexity of content available to users (Tian et al., 2024; Wang et al., 2024a). This surge in information has brought the critical task of fact-checking to the forefront (Huang et al., 2022). Standard approaches for automated fact-checking comprise three stages (Min et al., 2023; Chern et al., 2023; Wei et al., 2024; Setty and Setty, 2024): (1) breaking down the content into atomic claims[1]; (2) conducting web-based searches to gather relevant evidence; and (3) verifying each claim against the retrieved evidence. However, these methods struggle to effectively address certain types of issue, as illustrated in Figure 1.

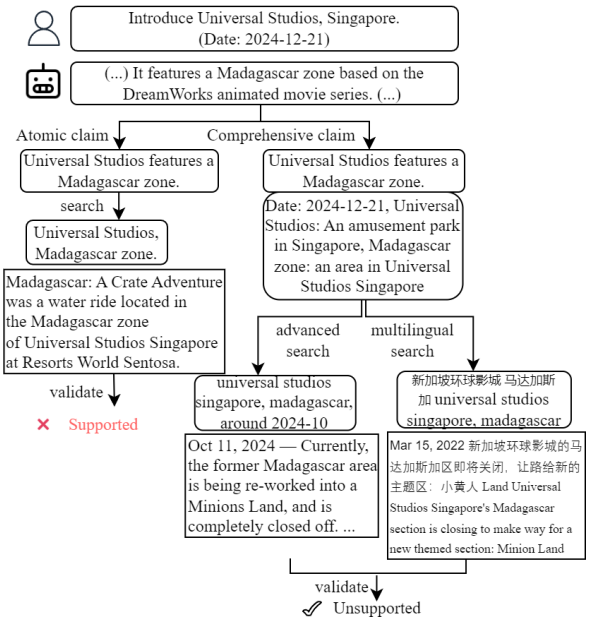

Figure 1: The workflow comparison between traditional fact-checking pipelines and PASS-FC, which enhances atomic claims by incorporating temporal and entity information, and utilizes advanced search and multilingual search to obtain relevant and sufficient evidence. The content within the rectangular boxes represents the retrieved evidence.

Firstly, the conventional definition of atomic facts overlooks the potential ambiguities in the claim date and entity references within the context of world knowledge (Gunjal and Durrett, 2024; Chiang and Lee, 2024). As illustrated in the left path of Figure 1, the Madagascar Zone at Universal Studios Singapore was once operational but had closed at the time of verification. This temporal nuance is missed by the atomic claim. The absence of precise temporal and entity specifications can lead to validation against evidence that, while seemingly relevant, does not accurately reflect the current state of affairs.

[1]An atomic claim is a short sentence conveying a single piece of information, as defined by Factscore (Min et al., 2023).

Secondly, the question generation module faces inherent challenges due to the uncertainty of retrieval outcomes (Lin et al., 2023). Search engines encompass vast and complex repositories of world knowledge, making it difficult to guarantee that a simple retrieval attempt based on existing information will yield relevant, credible, and sufficient evidence within the top results (Yu et al.; Song et al., 2024). This ambiguity is particularly pronounced in information-seeking scenarios, where users formulate queries about topics they are unfamiliar with (Park et al., 2021). The complexity of real-world information often requires multiple retrieval iterations (Khattab et al., 2021; Zhang et al., 2024) and sophisticated query refinement strategies to bridge the gap between initial queries and the desired evidence (Ousidhoum et al., 2022; Schlichtkrull et al., 2023; Chen et al., 2022). As illustrated in Figure 1, these strategies include the use of advanced search operators and multilingual retrieval techniques, which fully leverage search engine capabilities to enhance the efficiency and accuracy of the verification process.

In this paper, we propose PASS-FC, a framework that iteratively performs adaptive question generation and verification on decontextualized atomic facts. Specifically, PASS-FC first enhances the atomic facts by supplementing them with occurrence times and entity descriptions to prevent subsequent ambiguities (3.1.2). In the question generation module, PASS-FC incorporates advanced search and multilingual search capabilities, allowing the model to adjust its retrieval strategies based on the claim and previous feedback (3.2). Finally, the model reflects on the entire process and determines whether to initiate another round of retrieval, verification, or to conclude the process (3.5). Upon completion, PASS-FC returns a factuality label, providing a final assessment of the claim's veracity based on the comprehensive evidence gathered.

We conducted extensive testing on six diverse datasets, demonstrating that PASS-FC substantially improves fact verification performance, even outperforming models with stronger base capabilities in some scenarios. Our framework shows remarkable versatility, excelling in general knowledge, scientific, real-world, and multilingual fact-checking tasks (4.1). Furthermore, we performed a detailed analysis of various hyperparameters, revealing intriguing insights such as the optimal evidence count and the impact of reflection mechanisms. Our experiments uncovered that different language models benefit differently from iterative processes (4.2). Lastly, our ablation studies quantified the contribution of each module, demonstrating the crucial roles of claim augmentation, advanced search techniques, and language-specific adaptations in multilingual settings (4.3).

The main contributions of this paper are as follows: (1) We propose PASS-FC, a novel framework that enhances atomic facts with temporal and entity context, and employs iterative, adaptive question generation and verification for improved fact-checking accuracy. (2) We demonstrate PASS-FC's effectiveness across diverse fact-checking scenarios, including general knowledge, scientific, real-world, and multilingual tasks, often outperforming strong baseline models. (3) We provide insights into optimal fact-checking processes through comprehensive hyperparameter analysis and ablation studies, revealing the influence of evidence quantity, reflection mechanisms, and language-specific adaptations.

2 Related Work

Atomic Claim and Contextualization Effective fact verification requires a clear definition of facts (Ni et al., 2024), often necessitating the identification of atomic claims within longer texts. Numerous studies (Hu et al., 2024; Bayat et al., 2023; Min et al., 2023; Song et al., 2024) have proposed definitions for atomic facts. Chiang and Lee (2024) first highlighted the issue with atomic facts in which excessive atomization can lead to entity ambiguity. They addressed this problem through entity linking. Gunjal and Durrett (2024) introduced the concept of molecular facts, refining the approach by expanding entity and event descriptions within atomic facts. While these methods have shown promise, they have primarily been tested in biographical generation tasks. Our work distinguishes itself in several ways: Beyond supplementing entity descriptions, we also consider metadata such as temporal information that can contribute to ambiguity. Crucially, we extend beyond previous studies by evaluating the impact of these enhancements on fact verification in real-world scenarios.

Question Generation for Fact Verification To address real-time information demands (Fierro et al., 2024; Kasai et al., 2022) in real-world scenarios, many fact verification frameworks (Gao et al., 2022; Chen et al., 2023a; Sun et al., 2024) generate questions to retrieve relevant knowledge from

search engines like Google. Most existing works (Chern et al., 2023; Song et al., 2024; Wang et al., 2024a) rely on few-shot examples to prompt LLMs to ask direct questions or inquire about entity attributes in claims (Zhang et al., 2024; Wang and Shu, 2023). Schlichtkrull et al. (2023) demonstrates that human questioning strategies in fact verification are highly diverse, with an average similarity of only 0.25 between effective retrieval strategies across different individuals. Ousidhoum et al. (2022); Setty and Setty (2024) trained models to learn human-like questioning strategies based on existing datasets (Fan et al., 2020; Schlichtkrull et al., 2023; Chen et al., 2022; Park et al., 2021), but its domain is limited and lacks evaluation in real-world fact verification scenarios. Our work innovates by leveraging Google's advanced search operators[2] combined with multilingual retrieval, aiming to obtain more useful evidence.

Iterative Verification Verifying complex claims often requires multiple iterations to reach accurate conclusions. SAFE (Wei et al., 2024) and KnowHalu (Zhang et al., 2024) explicitly allow models to perform multiple retrievals until they deem all useful information has been obtained. Other approaches (Sun et al., 2024; Cohen et al., 2023) retrieve information only once but enable models to re-analyze verification results from different perspectives. ProgramFC (Pan et al., 2023) and Self-Checker (Li et al., 2024) treat all steps in the verification process as tools, allowing models to implicitly choose multiple retrievals or verifications by selecting appropriate tools at each step. Our work introduces a reflection step, which decides whether to retrieve more information, perform additional verification, or conclude the process after each verification cycle.

3 PASS-FC

We begin by formulating the fact-checking task. Given a query in any language, a response is generated either by a human or an LLM. The system's objective is to determine the veracity of this response within the context of the query and any relevant metadata, such as the date.

PASS-FC addresses this task through a two-step process, as illustrated in Figure 2. In the first step, the system extracts n comprehensive claims $\{c_1, c_2, \ldots, c_n\}$ from the response. Subsequently, for each claim, PASS-FC employs an iterative process of query generation, evidence retrieval, and claim verification. At the conclusion of each iteration, PASS-FC evaluates the process thus far, determining whether to terminate or continue. If continuation is warranted, it generates appropriate tools and instructions for the subsequent round. Prior to initiating the fact-checking process, PASS-FC implements a Language-Specific Initialization step. This phase involves configuring the model to think and operate in the source language specified by the user. By default, the source language is set to English, but it adapts to other languages based on user input.

Notably, this process operates independently of gold evidence. Each phase of PASS-FC is implemented through carefully crafted prompts to an LLM. Detailed example prompts are available in Appendix A.6. For psedocode of PASS-FC, please refer to algorithm 1.

3.1 Claim Detection

Claim detection is a crucial preliminary step in our fact-checking process, involving claim decomposition and augmentation. This step facilitates fact-checking by breaking down the response into suitable decontextualized atomic facts.

3.1.1 Claim Decomposition

Long-form responses often contain multiple pieces of information, making it impractical to fact-check them as a whole. Splitting the response into atomic facts has been widely recognized as an effective approach (Min et al., 2023; Chern et al., 2023). Drawing from recent NLP research (Ni et al., 2024; Gunjal and Durrett, 2024; Min et al., 2023), we define an atomic fact as a statement or assertion that: (1) can be objectively verified as true or false based on empirical evidence or reality, (2) represents a single, indivisible unit of information, (3) is self-contained and context-independent

Given an input response r with its corresponding prompt p, we decompose the response into a set of claims $\{c_1, c_2, \ldots, c_n\}$ that adhere to this definition. To ensure the atomicity of claims, we employ an iterative process: if a claim exceeds a predefined word threshold k[3], it undergoes further decomposition into simpler sub-claims. This approach strikes a balance between efficiency and effectiveness, resulting in verifiable, atomic, and self-contained claims.

[2] https://www.google.com/advanced_search

[3] Empirically predefined length threshold. Set to 10 for English.

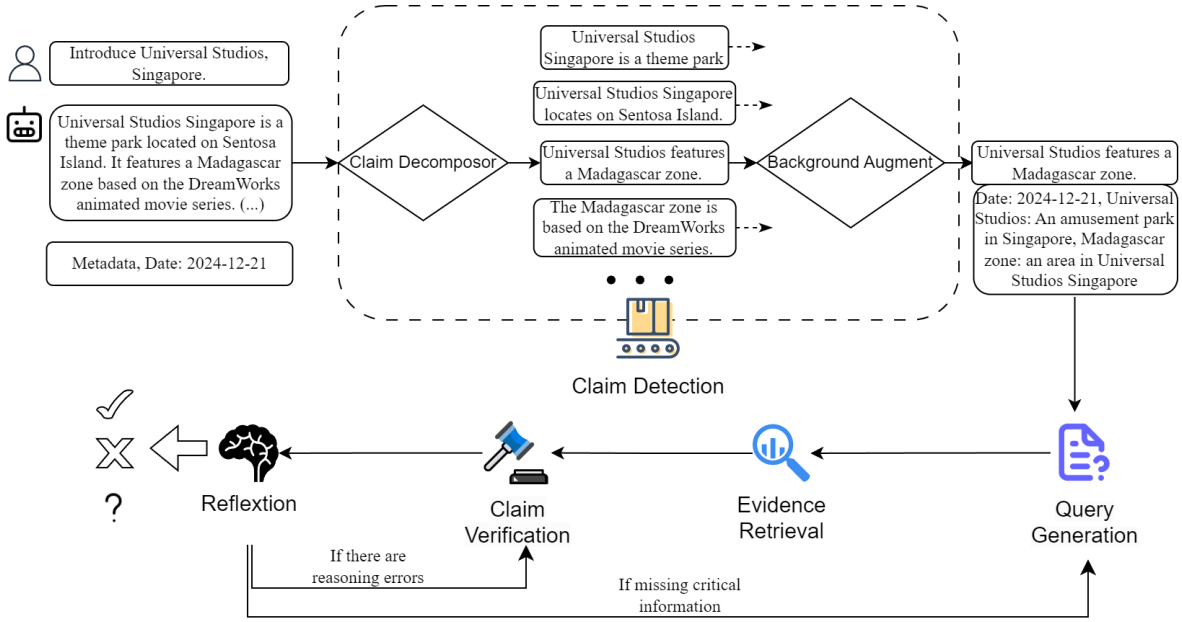

Figure 2: Overview of the PASS-FC framework.

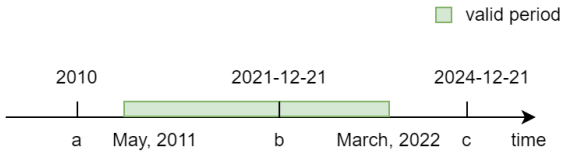

Figure 3: The timeline for the claim "Universal Studios features a Madagascar zone.". The valid period, shown as a green area, is marked with start and end dates from authorized sources. Three points (a, b, c) indicate claim dates inferred from the validation date (2024-12-21) and the claim's time description: a. "2010" b. "three years ago" c. No time description. If an inferred date falls outside the valid period, the claim is unsupported for that time description.

3.1.2 Claim Augmentation

Real-world claims often possess complex and nuanced contexts. Although we generate self-contained claims, they require decontextualization to avoid temporal and entity ambiguities in world knowledge. Claim augmentation addresses this by appending a claim period and entity brief to each atomic claim, forming a comprehensive claim as illustrated in Figure 2.

Claim Period We introduce the claim period to enhance the temporal context of factual claims. It represents the time span during which the event or situation described in the claim occurred or was true. To determine the claim period, our method analyzes explicit and implicit time references within the claim text, as well as metadata provided alongside the claim. Figure 3 demonstrates how variations in the claim's temporal description affect the corresponding claim period. A claim is considered true if its claim period falls within the time span covered by supporting evidence.

Entity Brief The entity brief is a crucial component of claim augmentation that focuses on uniquely specifying each entity referenced in a claim (Fan et al., 2020). Using the given prompt, response, and metadata, a language model reasons to provide concise descriptions for every entity mentioned. Unlike approaches such as Molecular Facts (Gunjal and Durrett, 2024), which only describe potentially ambiguous entities like person or place names, our method includes all entities in the brief. This comprehensive approach acknowledges the vast and dynamic nature of world knowledge, recognizing that entity ambiguity may only become apparent during the evidence-searching process. By fixing entity references within the fact-checking context, the entity brief ensures consistency throughout the verification process.

Figure 2 provides an example of how a claim is augmented with claim period and entity briefs, which work in tandem to identify relevant evidence pertaining to the same entities and time frame mentioned in the claim.

With the atomic claim augmented with a claim period and entity brief, PASS-FC proceeds to iteratively perform query generation, evidence retrieval,

claim verification, and reflection. The prompt, response, and metadata no longer serve as input context in the subsequent process. From this point forward, we refer to the augmented atomic claim as the comprehensive claim.

3.2 Query Generation

Our query generation module enhances the capability of LLMs to retrieve relevant, sufficient, and trustworthy evidence. This module processes a comprehensive claim along with any preceding step history to generate effective search queries. These queries are then used to extract evidence from search engines, facilitating the fact-verification process. The selection of specific search tools is determined by the reflection module based on previous history, with the advanced search tool being the default choice for the initial verification attempt.

3.2.1 Advanced Search

Advanced search operators are specialized commands and syntax employed by various search engines to refine and focus search results. These operators enable users to construct more precise queries by filtering information, specifying exact phrases, excluding certain terms, and combining multiple search criteria. Common examples include quotation marks for exact matches, Boolean operators (AND, OR), minus signs for exclusion, wildcards, and parentheses for grouping terms.

We incorporate an advanced search tool into our query generation module, leveraging these operators to create more targeted and effective search queries for fact verification. For each claim, our system generates two carefully crafted queries, utilizing appropriate operators to enhance the retrieval of relevant and reliable evidence. For a detailed explanation of the advanced search operators used in our system, please refer to Figure 8 in A.6, which contains the complete advanced search prompt.

3.2.2 Site-restricted Search

The site-restricted search tool is a crucial component of query generation module, designed to refine the scope of evidence retrieval. This tool enhances the fact-checking process in several ways. Firstly, it generates a list of credible domain suffixes for focused searching while identifying and excluding unreliable sources. This process leverages the LLM's extensive knowledge of domain names and incorporates insights from previous search results. Secondly, the tool offers flexibility by accommodating user input. Users can manually specify preferred domains (e.g., wiki.org), ensuring that all fact-checking evidence is sourced from a curated set of trusted websites. This feature allows for customized, domain-specific searches that align with user preferences.

The tool's output is seamlessly integrated with the advanced search queries, effectively narrowing the search to more reliable and relevant sources. Site-restricted search tool operates in parallel with the advanced search tool. While the advanced search tool can be used independently to generate queries without domain restrictions, the site-restricted search tool, when employed, works in conjunction with advanced search operators. This combination allows for more targeted queries that not only utilize advanced search techniques but also focus on specific, credible domains.

By integrating domain restrictions with advanced search operators, this approach directs searches to authoritative sources while maintaining the flexibility to perform broader searches when necessary. This targeted yet flexible method significantly enhances both the efficiency and accuracy of the fact-checking process, ensuring that the evidence gathered is not only relevant but also from credible sources.

3.2.3 Multilingual Search

To enhance the global applicability of PASS-FC, we introduce a multilingual search tool that expands the scope of evidence gathering beyond the user-defined source language. This tool analyzes the given claim to identify up to two relevant languages, in addition to the source language, from a predefined list of 46 languages supported by Google Search[4]. The source language, which defaults to English if not specified by the user, serves as the primary language for all steps in the fact-checking process.

The multilingual search tool considers key elements such as locations, people, news sources, and event places to determine which additional languages might offer more comprehensive, accurate, or up-to-date information. This approach is based on the principle that local language sources often provide more detailed and nuanced coverage of events or topics, particularly for claims involving international events, cultural phenomena, or region-

[4] https://support.google.com/googleplay/android-developer/table/4419860?hl=en&sjid=11904773475773808427-AP

specific topics.

Once these languages are identified, the system re-engages the advanced search query generation process, creating new queries in each of the selected languages. This crucial step ensures that the search terms are appropriately localized and culturally relevant, maximizing the effectiveness of the multilingual search.

3.3 Evidence Retrieval

Once queries are generated, they are submitted to a commercial search engine API[5] to collect relevant evidence. While multiple search engines support the advanced techniques we employ, we have chosen Google Search for its superior quality and comprehensive coverage. For each query, we retrieve the top-k (e.g., k=10) search results. From these results, we extract the title, snippet, and URL of each item. These elements are then combined to create a consolidated set of evidence for further analysis.

3.4 Claim Verification

The claim verification stage assesses whether a given claim is supported, contradicted, or inconclusive based on the available evidence. This process evaluates the consistency between the claim and the evidence, considering factual details, temporal aspects, and source credibility. The verification process is iterative, incorporating feedback from previous attempts to refine its judgment. We use three predefined veracity labels: supported (the majority of evidence corroborates the claim), contradicted (the majority of evidence opposes the claim), and inconclusive (either no relevant evidence is found, or there is conflicting evidence from credible sources that is temporally consistent with the claim).

3.5 Reflection

The reflection module analyzes the fact-checking process and determines whether further iteration is necessary. It provides textual feedback on the entire process and makes informed decisions based on historical steps and the current state.

When issues are identified, the module selects appropriate tools and generates feedback for improvement. If additional information is needed, it employs retrieval tools from the question generation module to refine search queries or explore specific credible domains. When key information has been overlooked, the module recommends revisiting the claim verification step, providing feedback that alters the model's reasoning approach.

The process concludes when the historical steps are deemed satisfactory or after a predetermined number of iterations. This adaptive mechanism ensures continuous improvement in fact-checking accuracy while maintaining efficiency.

3.6 History Management

Our history management approach adapts to the context length capabilities of different language models. For models with context lengths of 8,000 tokens or more, we maintain a dictionary containing all historical information, including brief descriptions and results of each step. For models with limited context lengths, such as GPT-3.5-Turbo (4,096 tokens), we retain only the comprehensive claim, the previous iteration's feedback, and the current round's history. This strategy optimizes performance across various language models while preserving essential contextual information.

4 Experiments

Datasets. To evaluate fact-checking capabilities across multiple domains, we selected six datasets for our experiments: FacTool-QA (N=233), FELM-WK (N=507), Factcheck-GPT (N=678), SciFact-Open (N=191), AVeriTec-Dev (N=500), and X-FACT (N=1,000) (Chern et al., 2023; Chen et al., 2023b; Wang et al., 2024a; Wadden et al., 2022; Schlichtkrull et al., 2023; Gupta and Srikumar, 2021). The first three datasets (FacTool-QA, FELM-WK, and Factcheck-GPT), collectively referred to as Factbench (hereafter), represent knowledge-based QA scenarios. This grouping, introduced in OpenFactCheck (Wang et al., 2024b), is based on their similar characteristics and domain focus. To be consistent with Wang et al. (2024b), We directly use the human generated atomic claims in Factbench, skipping the step of claim decomposition. SciFact-Open, AVeriTec-Dev, and X-FACT represent scientific, real-world, and multilingual fact-checking scenarios, respectively. Notably, AVeriTec and X-FACT contain the claim date as their metadata. For the other four datasets that lack this information, we used the first release date of the dataset's paper on arXiv as their claim date. The detailed dataset descriptions is provided in Appendix A.1.

Baselines. We included the following fact-checkers

[5] https://serper.dev/

Framework	Base Model	FacTool-QA F1	FacTool-QA Acc	FELM-WK F1	FELM-WK Acc	Factcheck-GPT F1	Factcheck-GPT Acc	SciFact-Open F1	SciFact-Open Acc	AVeriTeC F1	AVeriTeC Acc	X-FACT F1	X-FACT Acc	Average F1	Average Acc
FacTool	GPT-3.5-Turbo	0.625	0.650	0.560	0.606	0.635	0.686	0.796	0.796	0.582	0.570	0.569	0.572	0.628	0.647
Factcheck-GPT	GPT-4/GPT-4omini	0.755	**0.818**	**0.665**	**0.763**	**0.710**	0.674	0.588	0.560	0.576	0.562	0.316	0.246	0.602	0.604
SAFE	GPT-3.5-Turbo	0.591	0.627	0.465	0.466	0.572	0.550	0.676	0.686	0.610	0.630	0.412	0.449	0.554	0.568
PASS-FC	GPT-3.5-Turbo	0.672	0.785	0.586	0.704	0.698	**0.757**	0.728	0.733	0.666	0.658	**0.618**	**0.623**	0.661	0.710
PASS-FC	GPT-4omini	**0.770**	0.811	0.659	0.694	0.701	0.720	**0.858**	**0.859**	**0.692**	**0.694**	0.593	0.619	**0.712**	**0.733**

Table 1: Macro-F1 and accuracy for the fact-checking task. The highest results are in **bold**, and the second-best results are underlined. Results within the shaded area are cited from Wang et al. (2024b), where Factcheck-GPT uses GPT-4-Turbo as the base model. We ran the remaining Factcheck-GPT results with GPT-4omini to limit costs. Additionally, we tested the GPT-4o-powered PASS-FC on Factbench (see table 4), which significantly outperforms the baselines.

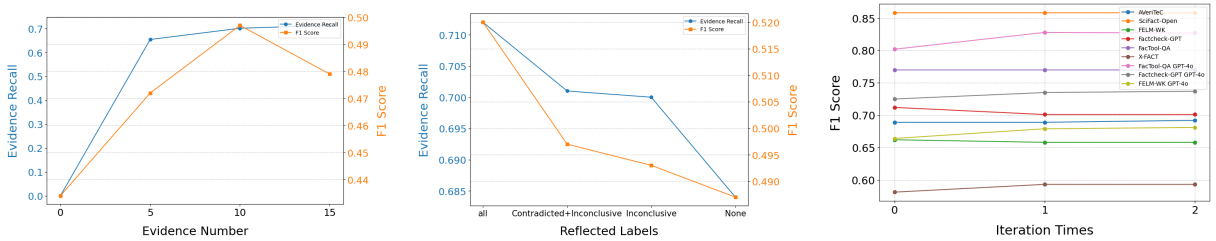

Figure 4: Hyperparameter analysis. (a) and (b): Impact of evidence number and reflection trigger labels on performance, using 100 randomly sampled AVeriTeC training examples. (c) Performance gains from iterations across all datasets using GPT-4omini, and on Factbench using GPT-4o. Evaluation metrics include evidence recall (where applicable) and macro-F1 score.

as baselines: FacTool, Factcheck-GPT, and SAFE, all using evidence retrieved from Google. These models were experimented using their default settings. More details about the models and their configurations can be found in Appendix A.2.

PASS-FC Setting. To focus on developing systems deployable across multiple downstream tasks, we refrained from additional hyperparameter tuning on the test datasets. We set English as the source language for all datasets except X-FACT, which uses its source language metadata. For all experiments, unless otherwise specified, we allowed all query generation tools, retrieved top-10 evidence for each query, set the maximum iteration number to 2, and had PASS-FC reflect only on Unsupported and Inconclusive labels. We used GPT-3.5-Turbo, GPT-4-Turbo, GPT-4o, and GPT-4omini in our experiments, with temperature sets to 0.

Metrics. Consistent with previous studies ((Wang and Shu, 2023; Li et al., 2024)), we use macro-F1 and accuracy as evaluation metrics. Macro-F1 averages the F1 scores of Supported and Unsupported labels, while accuracy is calculated across all label types. This approach slightly disadvantages PASS-FC, as it can output an Inconclusive label when uncertain, a category absent in most datasets used here (See Appendix A.3 for detailed discussion). Nevertheless, PASS-FC significantly outperforms all baselines in terms of average performance (Table 1).

4.1 Main Results

We report the overall results for PASS-FC and for the baselines for few-shot fact-checking in Table 1. We have two specific observations.

PASS-FC achieves competitive results with comparable or weaker base models. The experimental results show that PASS-FC, powered by GPT-3.5-Turbo, outperforms other baseline models using the same foundation model across various datasets. When equipped with GPT-4omini, PASS-FC exhibits performance comparable to Factcheck-GPT in the first three datasets and surpasses it in overall performance across all datasets. Additionally, table 4 demonstrate PASS-FC's strong performance on Factbench compared to baselines when equipped with GPT-4-Turbo or GPT-4o.

PASS-FC performs consistently across diverse fact-checking scenarios. The framework shows robust performance not only in the first three datasets focusing on general LLM knowledge question-answering but also in the latter three datasets representing scientific fact-checking, real-world scenarios, and multilingual fact verification. Notably, PASS-FC significantly outperforms baseline models on X-FACT and AVeriTeC datasets. This performance difference may be partially attributed to the consideration of claim timestamps, which baseline models overlook. We also show the reruned results for baselines in table 4, finding it's challenging to reproduce the original results in the shaded area of table 1. This discrepancy

Model	F1	Acc
PASS-FC	**0.672**	**0.785**
w/o Claim Augmentation	0.652	0.708
w/o Advanced Search	0.654	0.751
w/o Site-restricted Search	0.670	0.772
w/o Multilingual Search	0.672	0.785

Table 2: Ablation study results for PASS-FC on the FacTool-QA dataset.

Model	F1	Acc
PASS-FC	**0.593**	**0.619**
w/o Multilingual Search	0.574	0.605
w/o Language-Specific Initialization	0.571	0.599

Table 3: Ablation study results for PASS-FC on the X-FACT dataset.

likely stems from the absence of claim timestamps as metadata in the corresponding datasets, leading to verification based on the most recent information, which may have changed since the original claim was made. Further analysis of this aspect on fact-verification accuracy is in Appendix A.4

4.2 Configuration for the Best Performance

We conducted a comprehensive analysis of key hyperparameters in PASS-FC. Unless otherwise specified, all experiments used GPT-4omini as the base model with default settings. Figures 4 (a) and (b) explore the impact of evidence number and reflection trigger labels, respectively. To avoid overfitting on test data and leverage the gold evidence available, we randomly selected 100 examples from the AVeriTeC training dataset for these analyses. Figure 4 (a) reveals that as the number of retrieved evidence pieces increases, the evidence recall consistently improves. However, we observed that when the evidence count exceeds 10, the F1 score begins to decline. Figure 4 (b) demonstrates that expanding the set of labels that trigger reflection consistently leads to improved performance. This finding underscores the effectiveness of the reflection mechanism in enhancing fact-checking accuracy. Figure 4 (c) illustrates the impact of iteration count across all datasets. Interestingly, we found that unlike GPT-4o, GPT-4omini does not consistently show improvement with increased iterations. This observation aligns with the general understanding that GPT-4o possesses stronger capabilities, allowing it to benefit more from reflection.

4.3 Ablation Study

We conducted an ablation study to evaluate the contribution of each component in PASS-FC, using the FacTool-QA dataset with GPT-3.5-Turbo as the base model under default settings in table 2. Due to the omission of the claim decomposition step for FacTool-QA, we couldn't assess its impact. Notably, the claim augmentation, particularly the inclusion of the date metadata, significantly influenced performance. Even using the arXiv publication date ('2023-07-26') for FacTool claims, rather than individual claim verification dates, proved effective. This suggests that a considerable portion of facts in FacTool-QA may have changed over time. Replacing advanced search with FacTool's simpler direct questioning approach led to a performance decrease, indicating the value of advanced search even in general knowledge domains. The minimal impact of removing site-restricted search and multilingual retrieval can be attributed to FacTool-QA's focus on general domains, where English-based advanced search appears sufficient.

To further investigate the impact of PASS-FC components in a multilingual context, we conducted an additional ablation study on the X-FACT dataset using GPT-4omini in table 3. We tested two specific modifications: removing multilingual search and eliminating language-specific initialization (using English as the default source language regardless of input language). Both modifications led to a decrease in performance, with the removal of language-specific initialization having a slightly larger impact. These findings suggest that both multilingual search capabilities and language-specific initialization contribute to improved performance in multilingual fact verification tasks.

5 Conclusion

This paper presents PASS-FC, a novel fact-checking framework that uses iterative claim augmentation, advanced search, and multilingual capabilities to address challenges like temporal and entity ambiguities. Experiments show PASS-FC significantly improves fact verification across various datasets, outperforming stronger base models in general, scientific, real-world, and multilingual tasks. As online information evolves, PASS-FC will be vital for ensuring information integrity and supporting informed decisions.

Limitations

While it provides a reasonable definition of atomic facts, the framework only checks factual claims within the processed text. It is unable to detect truthful but irrelevant responses or evaluate model refusals to answer, limiting its scope in assessing overall response quality.

The framework relies solely on Google search snippets as evidence, without exploring full webpage content or following internal links. While PASS-FC is efficient, it potentially misses valuable information that could be obtained by deeper web exploration. Although such in-depth searching might significantly increase processing time, it could potentially improve fact-checking accuracy.

Preliminary observations suggest a possible antagonistic relationship between claim decomposition and iterative verification. This was noted in experiments where the performance gain from iterative verification was less pronounced on datasets like FactBench, which already incorporate claim decomposition. Further experiments are needed to verify this hypothesis and understand the interplay between these components.

The ablation studies are not comprehensive due to the lack of suitable benchmarks for testing individual modules. The framework's effectiveness is primarily judged by the final fact verification results, which may not fully reflect the performance of each component. Additionally, the advanced query generation tools proposed in this work are not triggered for every claim, necessitating the testing of a large number of examples to obtain statistically significant results.

Lastly, while using Google Search offers real-time advantages, it also introduces challenges in reproducing results. The constant updates to search results make it difficult to replicate exact experimental conditions, potentially affecting the consistency and comparability of fact-checking outcomes across different time periods.

References

Farima Fatahi Bayat, Kun Qian, Benjamin Han, Yisi Sang, Anton Belyi, Samira Khorshidi, Fei Wu, Ihab F. Ilyas, and Yunyao Li. 2023. Fleek: Factual error detection and correction with evidence retrieved from external knowledge. *Arxiv*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023a. Complex claim verification with evidence retrieved in the wild. NAACL 2024.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. EMNLP 2022.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I. Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023b. Felm: Benchmarking factuality evaluation of large language models. *Arxiv*.

I. Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Arxiv*.

Cheng-Han Chiang and Hung-yi Lee. 2024. Merging facts, crafting fallacies: Evaluating the contradictory nature of aggregated factual claims in long-form generations. *Arxiv*.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. Lm vs lm: Detecting factual errors via cross examination. *Arxiv*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs.

Constanza Fierro, Nicolas Garneau, Emanuele Bugliarello, Yova Kementchedjhieva, and Anders Sogaard. 2024. Mulan: A study of fact mutability in language models. *Arxiv*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2022. Rarr: Researching and revising what language models say, using language models. *Arxiv*.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in llm fact verification.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. *Preprint*, arXiv:2106.09248.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2024. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? 23 pages, 3 figures.

Kung-Hsiang Huang, ChengXiang Zhai, and Heng Ji. 2022. Concrete: Improving cross-lingual fact-checking with cross-lingual retrieval. Accepted by COLING 2022.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2022. Realtime qa: What's the answer right now? RealTime QA Website: https://realtimeqa.github.io/.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. NeurIPS 2021 (Spotlight).

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. 2024. Self-checker: Plug-and-play modules for fact-checking with large language models. *Arxiv*.

Kevin Lin, Kyle Lo, Joseph E. Gonzalez, and Dan Klein. 2023. Decomposing complex queries for tip-of-the-tongue retrieval.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *Arxiv*.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. Afacta: Assisting the annotation of factual claim detection with reliable llm annotators. *Arxiv*.

Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal question generation for fact-checking. Accepted at EMNLP 2022, 13 pages.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. *Arxiv*.

Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Faviq: Fact verification from information-seeking questions. ACL 2022(long). Data amp; Code available at https://faviq.github.io.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. Accepted to NeurIPS 2023 Datasets amp; Benchmarks Track.

Ritvik Setty and Vinay Setty. 2024. Questgen: Effectiveness of question generation methods for fact-checking applications. Accepted in CIKM 2024 as a short paper 4 pages and 1 page references. Fixed typo in author name; doi:10.1145/3627673.3679985.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. Veriscore: Evaluating the factuality of verifiable claims in long-form text generation.

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2024. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. 18 pages, 3 figures.

Jacob-Junqi Tian, Hao Yu, Yury Orlovskiy, Tyler Vergho, Mauricio Rivera, Mayank Goel, Zachary Yang, Jean-Francois Godbout, Reihaneh Rabbany, and Kellin Pelrine. 2024. Web retrieval agents for evidence-based misinformation detection. 1 main figure, 8 tables, 10 pages, 12 figures in Appendix, 7 tables in Appendix GitHub URL: https://github.com/ComplexData-MILA/webretrieval.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. Scifact-open: Towards open-domain scientific claim verification. *Preprint*, arXiv:2210.13777.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. *Arxiv*.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024a. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. *Arxiv*.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi Georgiev, Jiahui Geng, and Preslav Nakov. 2024b. Openfactcheck: A unified framework for factuality evaluation of llms. *Arxiv*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *Arxiv*.

Tian Yu, Shaolei Zhang, and Yang Feng. Auto-rag: Autonomous retrieval-augmented generation for large language models. Code is available at https://github.com/ictnlp/Auto-RAG.

Jiawei Zhang, Chejian Xu, Yu Gai, Freddy Lecue, Dawn Song, and Bo Li. 2024. Knowhalu: Hallucination detection via multi-form knowledge based factual checking. *Arxiv*.

A Appendix

A.1 Dataset Descriptions

Factool-QA The Factool-QA dataset (Chern et al., 2023) is a collection designed for fact-checking and question-answering tasks. It contains

50 real-world questions along with responses generated by ChatGPT. Each response is accompanied by annotated claims extracted from the AI-generated content. The questions in this dataset are sourced from various platforms, including Quora and TruthfulQA, providing a diverse range of topics. This dataset serves as a valuable resource for evaluating the accuracy of AI-generated responses and testing fact-checking systems.

FELM-WK The FELM-WK (World Knowledge) dataset (Chen et al., 2023b) is a subset of the larger FELM (Factuality Evaluation for Large language Models) collection, specifically focused on world knowledge. It contains 184 examples, which are further divided into 532 sub-claims. This dataset covers a wide range of topics including history, society, common sense, and current events. The questions in FELM-WK are sourced from various datasets and platforms such as TruthfulQA (Lin et al., 2022), Quora, hc3 (Guo et al., 2023), and MMLU (Lin et al., 2022), as well as some questions generated by ChatGPT and curated by the authors. FELM-WK is designed to evaluate the factual accuracy of language models in generating responses related to general world knowledge, making it a valuable resource for testing fact-checking systems and assessing the factual reliability of AI-generated content.

FactCheckGPT The FactCheckGPT dataset (Wang et al., 2024a) is a comprehensive collection designed for evaluating fact-checking systems and language models. It contains 184 examples, which are further divided into 532 sub-claims. The dataset focuses on fact-intensive content where language models are prone to hallucinate or produce factual errors. The examples are sourced from various origins, including ChatGPT-generated responses posted on Twitter, in-house brainstorming sessions, and selections from the dolly-15k dataset (Conover et al., 2023). The claims in FactCheckGPT are annotated for importance and checkworthiness, making it a valuable resource for testing fact-checking methodologies and assessing the factual accuracy of AI-generated content across a range of topics and complexities.

Scifact-Open The SciFact-Open dataset (Wadden et al., 2022) is distinct from previously mentioned datasets in that its claims are not generated by language models but are derived from scientific literature. In our study, we utilize a subset of 191 claims from SciFact-Open that have factuality labels. Unlike the original dataset design, which includes a corpus of abstracts for verification, our approach diverges by using Google Search for evidence retrieval instead of the provided evidence set. This modification allows us to test our fact-checking framework in a more open-ended, web-based setting. The SciFact-Open claims, being grounded in scientific literature, provide a rigorous benchmark for evaluating fact-checking systems on specialized, technical content, particularly in the domains of medicine and biology.

AVeriTeC The AVeriTeC (Automated VERIfication of TExtual Claims) dataset (Schlichtkrull et al., 2023) is a comprehensive resource for fact-checking research. In our study, we utilize the development set of AVeriTeC, as the test set labels are not publicly available. The dataset originally contains real-world claims annotated with question-answer pairs, veracity labels, and textual justifications. For our purposes, we modified the labeling scheme by merging the "not enough evidence" and "conflicting evidence/cherry-picking" categories into a single "inconclusive" label, aligning with our framework's classification approach. This adaptation of AVeriTeC provides a challenging benchmark for evaluating our fact-checking system on real-world claims, while maintaining a three-class labeling system (supported, refuted, inconclusive) consistent with our research objectives.

X-FACT The X-FACT dataset (Gupta and Srikumar, 2021), originally a multilingual fact-checking resource, has been adapted for our study. We selected claims labeled as either True or False, excluding other categories. From this subset, we chose 10 languages where both True and False labels had more than 50 instances each. These languages include English, Spanish, Italian, Indonesian, Polish, Portuguese, Romanian, Serbian, Turkish, and Russian. For each language, we randomly sampled 50 claims per label. The first eight languages are used from the training set, while Turkish and Russian serve as zero-shot test languages. This modification allows us to evaluate our fact-checking system's performance across diverse languages.

A.2 Baseline Descriptions

FacTool Factool (Chern et al., 2023) employs a tool-augmented approach for fact-checking. It operates in five stages: claim extraction using ChatGPT, query generation for each claim, evidence

collection via Google Search API, and agreement verification using ChatGPT or GPT-4. This framework integrates large language models with external tools to assess the factuality of claims, providing a comprehensive baseline for comparison with our PASS-FC model.

FactCheckGPT FactCheckGPT (Wang et al., 2024a) is a comprehensive baseline model that approaches fact-checking through a fine-grained, eight-step process. These steps include decomposition, decontextualization, checkworthiness identification, evidence retrieval and collection, stance detection, correction determination, claim correction, and final response revision. This structured approach allows for a detailed evaluation of each component in the fact-checking pipeline. While designed as a comprehensive framework, it offers flexibility in implementation, allowing for the combination of certain steps in practical applications. FactCheckGPT serves as a detailed comparison point for our PASS-FC model, providing insights into the performance of individual fact-checking subtasks.

SAFE SAFE (Search-Augmented Factuality Evaluator) (Wei et al., 2024) is a baseline model that employs a language model to evaluate the factuality of long-form responses. It operates in three main steps: (1) splitting the response into individual self-contained facts, (2) determining the relevance of each fact to the original prompt, and (3) verifying the factuality of relevant facts through iterative Google Search queries. SAFE's key innovation lies in its use of a language model to generate multi-step search queries and reason about the search results. The model outputs metrics including the number of supported, irrelevant, and not-supported facts. This approach provides a comprehensive factuality assessment, serving as a strong baseline for comparison with our PASS-FC model.

FactScore FActScore (Min et al., 2023) is a baseline model for fact-checking that first uses spaCy to segment long-form text into sentences, then decomposes these sentences into atomic facts using GPT-3.5. It employs various methods to verify these atomic facts, including a no-context language model approach, a retrieve-then-verify approach using Wikipedia, and a nonparametric probability method. FActScore calculates the overall factuality of a text by aggregating the verification results of its constituent atomic facts. This structured approach to fact-checking serves as a comparison point for our PASS-FC model in evaluating the factuality of comprehensive claims.

A.3 Label Standardization

Most test datasets have binary labels: supported and contradicted. FactCheckGPT and AVeriTeC include an additional 'inconclusive' label, but it only accounts for about 10% of the claims. Baseline models primarily output 'supported' and 'contradicted' labels. While SAFE can output an 'irrelevant' category, it's rarely used due to the factual nature of the test data. Thus a more favorable approach for baseline models was adopted. Macro-F1 is calculated as the average of F1 scores for 'supported' and 'contradicted' categories only. Accuracy is computed across all categories, providing a more balanced evaluation.

A.4 Claim Verification Results on Factbench

Table 4 demonstrate that PASS-FC consistently outperforms other frameworks when utilizing the same large language model. This superiority is particularly evident when comparing PASS-FC with GPT-4-Turbo to Factcheck-GPT with GPT-4-Turbo, which represents the best-performing baseline.

PASS-FC with GPT-4-Turbo achieves higher scores across all datasets and metrics compared to Factcheck-GPT with GPT-4-Turbo. This performance advantage is observed in the FacTool-QA, FELM-WK, and Factcheck-GPT datasets, as well as in the overall average scores.

The replication attempts for FacTool and Factcheck-GPT failed to reproduce the performance reported in the original studies. This discrepancy can be attributed to two main factors:

Firstly, the inherent variability and temporal evolution of search engine rankings contribute to inconsistencies in information retrieval. This variability introduces an element of randomness in the fact-checking process, potentially affecting the models' performance.

Secondly, and more significantly, these models do not account for the temporal context of claims, specifically the verification time metadata. The FacTool-QA dataset, for instance, lacks this crucial temporal information. PASS-FC addresses this limitation by inferring a verification time based on the dataset's publication date on arXiv (2023-07-26), assuming all claims were verified prior to this date.

Framework	Base Model	FacTool-QA		FELM-WK		Factcheck-GPT		Average	
		F1	Acc	F1	Acc	F1	Acc	F1	Acc
Perplexity.ai	Sonar-online	0.640	0.815	0.570	0.689	0.680	0.705	0.630	0.736
FactScore	GPT-3.5-Turbo	0.540	0.584	0.565	0.644	0.585	0.631	0.563	0.620
FacTool	GPT-3.5-Turbo	0.625	0.650	0.560	0.606	0.635	0.686	0.607	0.647
FacTool (Rerun)	GPT-3.5-Turbo	0.590	0.605	-	-	-	-	-	-
Factcheck-GPT	GPT-4-Turbo	0.755	0.818	0.665	**0.763**	0.710	0.674	0.710	0.751
Factcheck-GPT (Rerun)	GPT-4-Turbo	0.696	0.751	-	-	-	-	-	-
SAFE	GPT-3.5-Turbo	0.591	0.627	0.465	0.466	0.572	0.550	0.542	0.547
PASS-FC	GPT-3.5-Turbo	0.672	0.785	0.586	0.704	0.698	0.757	0.652	0.749
PASS-FC	GPT-4omini	0.770	0.811	0.659	0.694	0.701	0.720	0.710	0.742
PASS-FC	GPT-4o	**0.828**	**0.863**	0.681	0.708	**0.737**	0.743	**0.749**	0.771
PASS-FC	GPT-4-Turbo	0.810	0.854	**0.698**	0.752	0.734	**0.761**	0.748	**0.789**

Table 4: Macro-F1 and accuracy for the fact-checking task on the Factbench. The highest results are in **bold**, and the second-best results are underlined. Results within the shaded area are cited from OpenFactCheck ((Wang et al., 2024b)).

This temporal consideration is critical because the veracity of claims can change over time. A pertinent example from the FacTool-QA dataset illustrates this point: the claim "The United States has 94 operating reactors" may have been accurate at the time of dataset annotation in 2023. However, recent developments, such as the commencement of commercial operations at Vogtle Unit 4 in Georgia on April 29, 2024, have altered this fact[6].

Consequently, models evaluating this claim in 2024 without considering the original verification time might produce results that are factually correct for the current time but inconsistent with the dataset's ground truth labels. This temporal discrepancy underscores the importance of incorporating time-aware fact-checking mechanisms.

A.5 Pseudocode for PASS-FC Pipeline

A.6 Example Prompts

[6] https://www.eia.gov/tools/faqs/faq.php?id=228&t=21

Algorithm 1 PASS-FC Pipeline Algorithm

Require: Query q, Response r, Metadata m, Source Language l
Ensure: Veracity Label v

$claims \leftarrow$ `ClaimDetection`(q, r, m)
for each $claim \in claims$ **do**
  $c \leftarrow$ `ClaimAugmentation`$(claim)$
  $iteration \leftarrow 0$
  $evidence \leftarrow \emptyset$
  $history \leftarrow \emptyset$ {Initialize verification history}
  while $iteration <$ MAX_ITERATIONS **do**
    if need_new_queries **then**
      $queries \leftarrow$ `QueryGeneration`$(c, l, history)$ {History-aware query generation}
      $evidence \leftarrow$ `EvidenceRetrieval`$(queries)$
    end if
    $v \leftarrow$ `ClaimVerification`$(c, evidence, history)$ {History-informed verification}
    $(feedback, continue, need\_new\_queries) \leftarrow$ `Reflection`$(c, evidence, v, history)$
    $history \leftarrow history \cup \{(c, v, feedback, iteration)\}$ {Append complete context}
    if $\neg continue$ **then**
      break
    end if
    $iteration \leftarrow iteration + 1$
  end while
end for
return `AggregateLabels`$(history)$ {Final decision from history}

```
[Definitions about Fact]
Fact: A fact is a statement or assertion that can be objectively verified as true or false based on
     empirical evidence or reality.
Opinion: An opinion is a judgment based on facts, an attempt to draw a reasonable conclusion from factual
     evidence. While the underlying facts can be verified, the derived opinion remains subjective and is not
      universally verifiable.
Atomic Factual Claim: An atomic factual claim is a statement that explicitly presents one verifiable fact.
     Statements with subjective components like opinions can also contain factual claims if they explicitly
     present objectively verifiable facts.

[Instructions]
1. You are given a passage. Your task is to break the passage down into a list of atomic factual claims,
     based on the given [Definitions about Fact].
2. An atomic factual claim is a factual claim that cannot be decomposed. It only contains a singular piece
     of information.
3. Extract clear, unambiguous atomic factual claims to check from the input passage, avoiding vague
     references like 'he', 'she', 'it', or 'this', and using complete names.
4. Please accurately identify and extract every claim stated in the provided text. Each claim should be
     concise (less than 15 words).

[Input Format Instruction]
<context>: Context for <passage> to help you understand it better.
<passage>: The passage to extract claims from.

[Output Format Instruction]
1. Your response MUST be a list of dictionaries. Each dictionary should contains the key "claim", which
     correspond to the extracted claim (with all coreferences resolved).
2. You MUST only respond in the format as described below. DO NOT RESPOND WITH ANYTHING ELSE. ADDING ANY
     OTHER EXTRA NOTES THAT VIOLATE THE RESPONSE FORMAT IS BANNED. START YOUR RESPONSE WITH '['.

[response format]:
[
  {{
    "claim": "Ensure that the claim is fewer than 15 words and conveys a complete idea. Resolve any
         coreference (pronouns or other referring expressions) in the claim for clarity",
  }},
  ...
]

[Examples]
<context>: Who won the match between Tomas Berdych and Gael Monfis on Monte Carlo Masters, 2015?
<passage>: Tomas Berdych defeated Gael Monfis 6-1, 6-4 on Saturday. The sixth-seed reaches Monte Carlo
     Masters final for the first time. Berdych will face either Rafael Nadal or Novak Djokovic in the final.
<response>: [{{"claim": "Tomas Berdych defeated Gael Monfis 6-1, 6-4"}}, {{"claim": "Tomas Berdych defeated
     Gael Monfis 6-1, 6-4 on Saturday"}}, {{"claim": "Tomas Berdych reaches Monte Carlo Masters final"}},
     {{"claim": "Tomas Berdych is the sixth-seed"}}, {{"claim": "Tomas Berdych reaches Monte Carlo Masters
     final for the first time"}}, {{"claim": "Berdych will face either Rafael Nadal or Novak Djokovic"}},
     {{"claim": "Berdych will face either Rafael Nadal or Novak Djokovic in the final"}}]

<context>: How many photos does Tinder show by default, and can users access additional photos beyond this
     limit?
<passage>: Tinder only displays the last 34 photos - but users can easily see more. Firm also said it had
     improved its mutual friends feature.
<response>: [{{"claim": "Tinder only displays the last photos"}}, {{"claim": "Tinder only displays the last
     34 photos"}}, {{"claim": "Tinder users can easily see more photos"}}, {{"claim": "Tinder said it had
     improved its feature"}}, {{"claim": "Tinder said it had improved its mutual friends feature"}}]

Now complete the following, ONLY RESPONSE IN A LIST FORMAT, NO OTHER WORDS!!!:
<context>: {prompt}
<passage>: {input}
<response>:
```

Figure 5: The prompt used in Claim Decomposition.

```
[Definitions about Fact]
Fact: A fact is a statement or assertion that can be objectively verified as true or false based on
     empirical evidence or reality.
Opinion: An opinion is a judgment based on facts, an attempt to draw a reasonable conclusion from factual
     evidence. While the underlying facts can be verified, the derived opinion remains subjective and is not
      universally verifiable.
Atomic Factual Claim: An atomic factual claim is a statement that explicitly presents one verifiable fact.
     Statements with subjective components like opinions can also contain factual claims if they explicitly
     present objectively verifiable facts.

[Instructions]
1. You are given a passage, a factual claim extracted from the passage, and a text feedback about the
     problem of the extracted claim. If the claim can be completely decomposed into multiple atomic factual
     claims, your task is to break the given factual claim down into a list of atomic factual claims, based
     on the given [Definitions about Fact]. Otherwise, return "None".
2. An atomic factual claim is a factual claim that cannot be decomposed. It only contains a singular piece
     of information.
3. If the feedback contains rational suggestions, they should be adopted to refine the final result.
4. Extract clear, unambiguous atomic factual claims to check from the given claim, avoiding vague references
      like 'he', 'she', 'it', or 'this', and using complete names.
5. Please accurately identify and extract every claim stated in the provided text. Each claim should be
     concise (less than 15 words).

[Input Format Instruction]
<context>: Context for <passage> to help you understand it better.
<passage>: The passage where the following claim is extracted from.
<extracted_claim>: A claim that was extracted from the passage in a former round of claim extraction.
<feedback>: Feedback about the former round of claim extraction. It may contain problems about the extracted
      claim and corresponding suggestions.


[Output Format Instruction]
1. If the <extracted_claim> is already an atomic factual claim which cannot be further decomposed, output "
     None".
2. Otherwise, your response MUST be a list of dictionaries. Each dictionary should contains the key "claim",
      which correspond to the extracted claim (with all coreferences resolved).
3. You MUST only respond in the format as described below. DO NOT RESPOND WITH ANYTHING ELSE. ADDING ANY
     OTHER EXTRA NOTES THAT VIOLATE THE RESPONSE FORMAT IS BANNED. START YOUR RESPONSE WITH '['.

[response format]
[
  {{
    "claim": "Ensure that the claim is fewer than 15 words and conveys a complete idea. Resolve any
         coreference (pronouns or other referring expressions) in the claim for clarity",
  }},
  ...
]

[Examples]
{examples}

Now complete the following, ONLY RESPONSE IN A LIST OF DICTIONARIES FORMAT, NO OTHER WORDS!!!:
<context>: {prompt}
<passage>: {input}
<extracted_claim>: {claim}
<feedback>: {feedback}
<response>:
```

Figure 6: The prompt used in Reflected Claim Decomposition.

```
[Definitions]
1. Fact: A fact is a statement or assertion that can be objectively verified as true or false based on empirical evidence or
      reality.
2. Atomic Factual Claim: An atomic factual claim is a statement that explicitly presents one verifiable fact. Statements with
      subjective components like opinions can also contain factual claims if they explicitly present objectively verifiable
      facts.
3. Named Entity: A named entity is a real-world object, such as a person, location, organization, product, etc., that can be
      denoted with a proper name. It is a phrase that uniquely refers to an object by its proper name, acronym, or
      abbreviation.
4. Vague references are words or phrases that do not clearly specify their subject. These references may be clear in the
      original context but become ambiguous when the claim is isolated. Vague references include but are not limited to:
  - Pronouns (e.g., "his", "they", "her")
  - Unknown entities (e.g., "this event", "the research", "the invention")
  - Non-full names (e.g., "Jeff..." or "Bezos..." when referring to Jeff Bezos)

[Instructions]
1. You are given a <CLAIM> and its broader context, which includes a <PROMPT>, the <RESPONSE> to that prompt, and additional
      background information. The <CLAIM> is extracted from the <RESPONSE>, obeying the definiton of "Atomic Factual Claim"
      mentioned before.
2. Based on the given [Definitions], you need to first resolve vague references in the <CLAIM>, then augment the revised claim
       with its Time, and Named Entity information, ensuring each attribute helps to uniquely identify the fact and its
      context.
3. Requests for resolving vague references:
  a. Identify any vague references in the <CLAIM>.
  b. Replace these vague references with proper entities from the <RESPONSE> or context.
  c. Do not change any factual claims or add new information.
4. After resolving vague references, augment the revised <CLAIM> with the following background attributes based strictly on
      the information provided in the revised <CLAIM> and its context:
  a. Time: Specify the time when the fact in the claim holds true, based solely on the description in the revised <CLAIM> and
        its context. The "time" key represents the temporal context or validity period of the claim. It indicates when the
        statement is or was true, or from which point in time the information holds. This is crucial for facts that can change
        , such as political positions or current events. If there's no explicit time description in the claim or context, use
        "Now" as the default, indicating the fact is assumed to be true at present. Brief steps are:
    - If explicitly stated, use that time.
    - If not stated but implied, infer from context.
    - If no time information, use "Now".
  b. Entity: List named entities mentioned in the claim, providing brief but distinguishing descriptions based only on
        information given in the claim or context. It's because one named entity can refer to multiple objects. For instance,
        the city "Birmingham" could be "Birmingham, Alabama, USA" or "Birmingham, West Midlands, UK". Do not add any
        information that isn't explicitly stated or directly implied. Brief steps are:
    - List each entity in the claim.
    - Provide brief descriptions using only information from the claim or context.
5. Remember to maintain the original meaning of the claim while making it more precise and informative. The goal is to create
      a claim that is unambiguous and can be understood correctly even without additional context.
6. Before giving your revised statement, think step-by-step and show your reasoning.

[Input Format Instruction]
<PROMPT>: Context for <RESPONSE> to help you understand it better.
<RESPONSE>: The passage where the following claim is extracted from. It's also the response of the former <PROMPT>.
<CLAIM>: The claim that was extracted from the <RESPONSE>.

[Output Format Instruction]
You should only respond in format as described below. DO NOT RETURN ANYTHING ELSE. START YOUR RESPONSE WITH '{{'.
[response format]:
{{
  "reasoning": "Find each vague reference. Explain what each vague reference likely refers to based on the claim and context.
        Show how you arrived at each conclusion. Then explain your reasoning for the Time attribute. Finally describe how you
        determined the Entity information.",
  "revised_claim": "Output the resolved claim.",
  "time": "The time when the fact in the claim holds true, using only information from the given claim and context.",
  ...
}}


[Examples]
<PROMPT>: You are a travel assistant. I will give you some reference documents of Singapore. Please output "Singapore's
      attraction introduction, mainly introducing the characteristics of attractions and what can be done here", output
      language must be English.
<RESPONSE>: Universal Studios Singapore, located within Resorts World Sentosa, is a cinematic adventure park that brings the
      silver screen to life with its thrilling rides and attractions . Each of its six themed zones offers a unique experience
      , from the prehistoric landscapes of The Lost World to the enchanting realm of Far Far Away . Visitors can immerse
      themselves in the futuristic Sci-Fi City, explore the mysteries of Ancient Egypt, or feel the buzz of New York and
      Hollywood's iconic streets . With 24 rides and attractions, including adrenaline-pumping roller coasters like Battlestar
       Galactica and family-friendly experiences such as the Madagascar river boat journey, there's something for every age
      and level of adventure . Live shows, character meet-and-greets, and a variety of dining and shopping options enhance the
       park's appeal, making it a must-visit destination for movie enthusiasts and thrill-seekers alike.
<CLAIM>: Universal Studios Singapore has six themed zones.
<OUTPUT>: {{"reasoning": "The subject in the claim is \"Universal Studios Singapore\", which is not a pronoun and does not
      reference an unknown entity. Furthermore, \"Universal Studios Singapore\" is not further specified in the RESPONSE, so
      we can assume that it is a full name. Therefore, there are not any vague references in the claim. The context did not
      include any specific time for its description. By default, we believe the RESPONSE still holds \"Now\". The entity \"
      Universal Studios Singapore\" need to be specified to avoid ambiguity.", "revised_claim": "Universal Studios Singapore
      has six themed zones.", "time": "Now", "Universal Studios Singapore": "located within Resorts World Sentosa, Singapore
      "}}


Now complete the following, ONLY RESPONSE IN A DICT FORMAT, NO OTHER WORDS!!!:
<PROMPT>: {prompt}
<RESPONSE>: {response}
<CLAIM>: {claim}
<OUTPUT>:
```

Figure 7: The prompt used in Claim Augmentation.

```
[Definitions]
1. Google advanced search operators are special commands and characters that filter search results.
2. Fact: A fact is a statement or assertion that can be objectively verified as true or false based on
     empirical evidence or reality.

[Google Advanced Search Operators]
| Search operator | What it does                                                | Example                  |
|-----------------|-------------------------------------------------------------|--------------------------|
| " "             | Put any phrase in quotes to force Google to use exact-match. On single words, prevents
     synonyms. | "nikola tesla"      |
| OR              | Google search defaults to logical AND between terms. Specify "OR" for a logical OR (ALL-
     CAPS). | tesla OR edison           |
| -               | Put minus (-) in front of any term (including operators) to exclude that term from the
     results. | tesla -motors       |
| *               | An asterisk (*) acts as a wild-card and will match on any word. | tesla "rock * roll"
          |
| ( )             | Use parentheses to group operators and control the order in which they execute. |(tesla
     OR edison) alternating current|
| before:         | Search for results from before a particular date. | apple before:2007-06-29  |
| after:          | Search for results from after a particular date.  | apple after:2007-06-29  |
| loc:            | Find results from a given area.                   | loc:"san francisco" apple |

[Instructions]
1. You and your partners are on a mission to fact-check a paragraph. Subclaims requiring verification have
     been extracted from the paragraph. Imagine yourself as an internet research expert. Your task is to
     generate two search queries for the provided claim to find relevant information for fact-checking.
     Please ensure that all queries are direct, clear, and explicitly relate to the specific context
     provided in the question and answer.
2. Utilize advanced Google search techniques when appropriate. But do not use site operators (e.g., site:
     example.com) in your queries, even if suggested in the feedback. Another tool will handle domain-
     specific searches separately.
3. Some searches have already been performed on this <CLAIM>. Please also consider the historical search
     information <HISTORY>. Adjust the queries based on the feedback from previous searches, focusing on
     areas where evidence was lacking or unclear.
4. Use date-based or location-based searches (before, after, and loc) only if: a) Historical search
     information is provided, AND b) The feedback in <HISTORY> explicitly indicates that the current search
     results are not within the required date range or destination.

[Output Format Instruction]
You should only respond in format as described below (a Python list of queries). PLEASE STRICTLY FOLLOW THE
     FORMAT. DO NOT RETURN ANYTHING ELSE. START YOUR RESPONSE WITH '['.
[response format]: ['query1', 'query2']

[Examples]
Here are three examples:
<CLAIM>: Michael Phelps is the most decorated Olympian of all time.
<RESPONSE>: ["Who is the most decorated Olympian of all time?", "Michael Phelps"]

<CLAIM>: Tesla is an American rock band formed in 1984.
<RESPONSE>: ["When is the rock band tesla formed?", "Rock band tesla Introduction. -motors -car -battery"]

<CLAIM>: Apple is used in various culinary applications. (The fruit apple)
<RESPONSE>: ["Apple's application in culinary. -phone -company", "Cooking ways of apple."]

Now complete the following(ONLY RESPONSE IN A LIST FORMAT, DO NOT RETURN OTHER WORDS!!! START YOUR RESPONSE
     WITH '[' AND END WITH ']'):
<CLAIM>: {input}
<HISTORY>: {feedback}
<RESPONSE>:
```

Figure 8: The prompt used in Advanced Search.

```
[Instructions]
1. You are an AI assistant tasked with verifying the truthfulness of a given claim. Your goal is to provide
     domain names of potentially relevant, credible, and authoritative sources while excluding unreliable
     sources.
2. Only provide domain suffixes, not full URLs.
3. Include reliable sources such as:
  Government and official websites (.gov, .org)
  Encyclopedia websites (.wiki)
  Reputable news outlets (provide their official domain names)
4. Exclude unreliable sources like personal comments from forums or social media platforms.
5. If provided with a history of previous actions, which may include past searches and feedback. Focus on
     the search results and feedback.
  If official sources were found but didn't provide sufficient information, include them in your output for
       targeted searching
  If personal comments from forums were found, exclude those domains (mark with a minus sign, e.g., -reddit.
       com)
6. In summary, for each claim, provide:
  Recommended domain suffixes for searching
  Domains to exclude (marked with a minus sign)
  Any official sources from previous searches that warrant further investigation
7. You should only respond in format as described below (a Python list). PLEASE STRICTLY FOLLOW THE FORMAT.
     DO NOT RETURN ANYTHING ELSE. START YOUR RESPONSE WITH '['.
[response format]: ['url1', 'url2', '-url3']

[Examples]
<CLAIM>: The Eiffel Tower was built in 1889.
<RESPONSE>: ['.gov.fr', '.paris.fr', '.unesco.org', .'britannica.com', '-tripadvisor.com', '-reddit.com']

<CLAIM>: COVID-19 vaccines are safe and effective.
<RESPONSE>: ['.who.int', '.cdc.gov', '.nih.gov', '.edu', '-facebook.com', '-twitter.com']

<CLAIM>: Global temperatures have risen significantly in the past century.
<RESPONSE>: ['.nasa.gov', '.noaa.gov', '.ipcc.ch', '.nature.com', '-climatechangehoax.com', '-blogspot.com']

Now complete the following(ONLY RESPONSE IN A LIST FORMAT, DO NOT RETURN OTHER WORDS!!! START YOUR RESPONSE
     WITH '[' AND END WITH ']'):
<CLAIM>: {input}
<HISTORY>: {feedback}
<RESPONSE>:
```

Figure 9: The prompt used in Site-Restricted Search.

```
[Supported Languages]
["Afrikaans", "Amharic", "Bulgarian", "Catalan", "Chinese (Hong Kong)",
"Chinese (PRC)", "Chinese (Taiwan)", "Croatian", "Czech", "Danish",
"Dutch", "Estonian", "Filipino", "Finnish", "French (Canada)",
"French (France)", "German", "Greek", "Hebrew", "Hindi", "Hungarian", "Icelandic", "Indonesian", "Italian",
     "Japanese", "Korean", "Latvian",
"Lithuanian", "Malay", "Norwegian", "Polish", "Portuguese (Brazil)",
"Portuguese (Portugal)", "Romanian", "Russian", "Serbian", "Slovak",
"Slovenian", "Spanish (Latin America)", "Spanish (Spain)", "Swahili",
"Swedish", "Thai", "Turkish", "Ukrainian", "Vietnamese", "Zulu"]

[Instructions]
1. You are an AI assistant tasked with analyzing claims and determining the most appropriate languages for
     fact-checking and evidence gathering. Your goal is to identify languages, other than English, that
     might provide more accurate, detailed, up-to-date, and factual evidence for a given claim.
2. When presented with a claim, analyze it for key elements such as locations, people, news sources, and
     event places. Based on these elements, determine if there are countries whose official languages might
     offer better sources of information. Consider that local languages often provide more detailed and
     accurate information.
3. If you identify relevant languages other than English, select up to two languages from the provided list
     that are most likely to yield valuable information. If only English is deemed suitable, do not output
     any languages, direct output None.
4. You should only respond in format as described below (a Python list of languages). Output None if only
     English is deemed suitable. PLEASE STRICTLY FOLLOW THE FORMAT. DO NOT RETURN ANYTHING ELSE. START YOUR
     RESPONSE WITH '['.
[response format]: ['language1', 'language2']

[Examples]
<CLAIM>: Angela Merkel announced her retirement from politics in 2021.
<RESPONSE>: ["German"]

<CLAIM>: Samsung unveiled its latest foldable smartphone at an event in Seoul.
<RESPONSE>: ["Korean"]

<CLAIM>: The 2024 Carnival in Rio de Janeiro is expected to be the largest in history.
<RESPONSE>: ["Portuguese (Brazil)"]

<CLAIM>: Tensions between Russia and Ukraine escalated after the incident in the Kerch Strait.
<RESPONSE>: ["Russian", "Ukrainian"]

<CLAIM>: NASA's Perseverance rover discovered new evidence of ancient microbial life on Mars.
<RESPONSE>: None

Remember, the goal is to enhance fact-checking by identifying languages that might provide more
     comprehensive or accurate information than what's available in English sources alone.

Now complete the following:
<CLAIM>: {input}
<RESPONSE>:
```

Figure 10: The prompt used in Multilingual Search.

```
You are given a piece of factual claim. Your task is to identify whether there are any factual errors within
     the claim. Besides a claim, background information about the claim such as its valid time and
     descriptions of potentially confusing named entities is also provided. Note that the background
     information is only for better understanding of the claim. You SHOULD NOT judge the factuality of the
     background information.

Some evidence has already been retrieved in previous attempts to verify the claim, and several(at least one)
     verifications were made. The previous series of actions taken are saved in <STEP_HISTORY>. However,
     upon further reflection, it was discovered that there were issues with the previous verification
     process, and negative <FEEDBACK> was provided. The <FEEDBACK> includes an analysis of the problems with
     the previous attempt and a plan for the direction of future steps.
Based on the <FEEDBACK>, additional evidence may have been retrieved. If provided, please make good use of
     this <NEW_EVIDENCE>, along with the previous verification process stored in <STEP_HISTORY>, to verify
     the claim. If only the <FEEDBACK> is provided without new evidence, it means that the existing evidence
     in <STEP_HISTORY> is sufficient. You should follow the instruction in the <FEEDBACK> to identify and
     consider relevant evidence for re-verification.

When you are judging the factuality of the given text, you could reference the provided evidences if needed.
     The provided evidences may be helpful. You must be careful when using the evidence to judge the
     factuality of the given text. Supportive evidence should be consistent with the claim in terms of the
     facts, time, and named entities. Some evidence may contradict each other. But good evidence should
     comes from a credible source. If there is insufficient evidence to either support or refute the claim,
     classify it as INCONCLUSIVE.
The response should be a dictionary with four keys - "reasoning", "factuality", "error", and "correction",
     which correspond to the reasoning, whether the given text is factual or not (Boolean - True, False, or
     INCONCLUSIVE), the factual error present in the text, and the corrected text.

Now complete the following, ONLY RESPONSE IN A DICT FORMAT, NO OTHER WORDS!!!:
<CLAIM>:
{claim}
<STEP_HISTORY>:
{steps}
<FEEDBACK>:
{feedback}
<NEW_EVIDENCE>:
{evidence}
<OUTPUT>:
```

Figure 11: The prompt used in Claim Verification.