FIRE 🔥: Fact-checking with Iterative Retrieval and Verification

Anonymous ACL submission

Abstract

Fact-checking long-form text is challenging, and it is therefore common practice to break it down into multiple atomic claims. The typical approach to fact-checking these atomic claims involves retrieving a fixed number of pieces of evidence, followed by a verification step. However, this method is usually not cost-effective, as it underutilizes the verification model's internal knowledge of the claim and fails to replicate the iterative reasoning process in human search strategies. To address these limitations, we propose FIRE, a novel agent-based framework that integrates evidence retrieval and claim verification in an iterative manner. Specifically, FIRE employs a unified mechanism to decide whether to provide a final answer or generate a subsequent search query, based on its confidence in the current judgment. We compare FIRE with other strong fact-checking frameworks and find that it achieves slightly better performance while reducing large language model (LLM) costs by an average of 7.6 times and search costs by 16.5 times. These results indicate that FIRE holds promise for application in large-scale fact-checking operations.

1 Introduction

> *"Every man has a right to his opinion, but no man has a right to be wrong in his facts."* - Bernard M. Baruch

Large language models (LLMs) have demonstrated exceptional performance across a wide range of tasks, including both language comprehension and generation (Zhao et al., 2023; Xie et al., 2023a). Consequently, LLMs are now widely applied in various domains (Xie et al., 2023b), and many users increasingly rely on the information they provide. However, this reliance is problematic, as LLMs are capable of producing outputs that are highly confident but factually incorrect, highlighting the critical need for robust fact-checking systems (Akhtar et al., 2023). However, fact-checking the entire output of LLMs in a single step is highly challenging. To address this, Min et al. (2023) proposed decomposing the content into multiple atomic claims, each of which can be individually verified. While this approach simplifies the fact-checking process, assessing the veracity of these atomic claims remains complex, especially when many require sourcing evidence from the web. Indeed, identifying the most relevant evidence online is a key challenge in fact-checking pipelines (Wang et al., 2023).

To address this issue, conventional methods, such as FACTOOL and FACTCHECK-GPT (Chern et al., 2023; Wang et al., 2023), frame the problem as a question-answering task, as illustrated on the left side of Figure 1. In these approaches, an LLM is prompted to generate N relevant questions, which are then used as search queries by a web search tool. The search results serve as evidence for LLM to determine the factuality of the claim. However, we argue that this process is inefficient in two key aspects. First, it underutilizes the internal knowledge already embedded in LLMs during pretraining. For claims involving common knowledge or widely known events, the LLM could confidently assess the claim without relying on external information. Second, generating multiple search queries concurrently does not align with the typical human reasoning process during search (Hu et al., 2023). Humans tend to begin with an initial query, gather information, and then refine their perspective on the claim, which often leads to the formulation of more effective follow-up queries.

To address this gap, we introduce **F**act-checking with **I**terative **R**etrieval and V**E**rification (FIRE), an innovative agent-based framework that integrates both the internal knowledge of LLMs and external knowledge sources by unifying the verification process and search query generation into a single step. As illustrated on the right side of Figure 1, FIRE employs a mechanism to decide

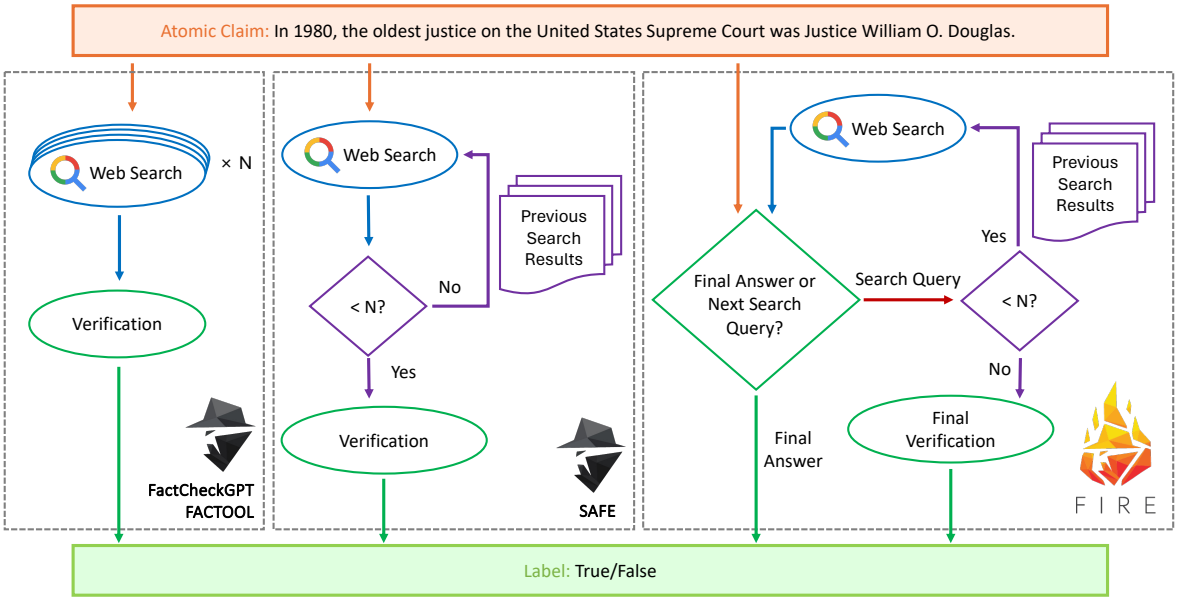

Figure 1: **Comparisons between FIRE and previous frameworks.** Previous frameworks typically treat web search and claim verification as distinct processes. In contrast, FIRE integrates interactive retrieval and verification.

whether to produce the final answer or generate a new search query, continuing the evidence-seeking process. This decision is based on the model's confidence in its judgment. The closest related work to us is SAFE (Wei et al., 2024), depicted in the center of Figure 1. Their method generates web search queries iteratively and subsequently verifies whether the entire retrieved evidence supports the claim. However, this approach lacks flexibility, as it treats evidence retrieval and claim verification as distinct processes, requiring a predetermined fixed number of searches regardless of the claim's complexity. In contrast, our approach integrates evidence retrieval and claim verification into an iterative framework, encouraging the language model to verify based on its own knowledge and conduct searches only when necessary. **Our experiments demonstrate that our method significantly reduces the computational costs of LLMs by an average factor of 7.6, as well as search-related costs by a factor of 16.5, all while maintaining fact-checking performance.**

In summary, our contributions are as follows:

- We present FIRE, a simple yet effective interactive framework for fact-checking. Through extensive experiments conducted across multiple datasets, we demonstrate that our framework significantly reduces the LLM computational and search costs, making it a better option for large-scale production.
- Our ablation studies demonstrate that the step-by-step reasoning process enhances the model's confidence in fact-checking, particularly with GPT-4o-mini. For GPT-4o, we observed a similar trend; however, the effect was not as pronounced as that seen with GPT-4o-mini.
- We conducted an error analysis and identified several quality issues in the current benchmark datasets, including the presence of ungrounded claims. Additionally, the strict reasoning capabilities of the LLM may incorrectly classify some debatable claims as non-factual.

2 Related Work

LLM Factuality Despite the remarkable capabilities of LLMs (Brown et al., 2020; OpenAI, 2023; Zhao et al., 2023), the auto-regressive learning objective does not inherently offer strong guarantee or enforce the learning of factual accuracy in the training process, making these models produce content that deviates from real-world facts (Wang et al., 2024a). On average, there are 5%-10% false claims in responses of GPT-4 (OpenAI, 2023) and LLaMA-2 (Touvron et al., 2023) on world-knowledge questions (Iqbal et al., 2024). Retrieval-augmented generation (Guu et al., 2020) and post-generation fact-checking are essential for ensuring accurate knowledge dissemination. Retrieving

highly relevant information plays a pivotal role in both guiding generation as a reference and determining verification results in fact-checking systems (Wang et al., 2023).

The retriever and verifier are the most resource-consuming components in fact-checking systems, in terms of time and cost. Even with the inexpensive APIs (e.g., Serper at 0.001 USD per request and GPT-3.5-turbo for verification), verifying an atomic claim costs approximately 0.02 USD, making extensive verification impractical for general users (Iqbal et al., 2024). This high cost limits the ability to verify large volumes of LLM responses, potentially contributing to the spread of misinformation. Our framework aims to minimize the costs in these two steps, enabling affordable verification for general users. This allows them to easily verify suspicious or doubtful information, enhancing the dissemination of factual information.

Fact Checking with Agents The recent advancements in LLMs have spurred significant research on LLM-powered agents, which are capable of reasoning about their environment and making decisions by either invoking external tools or performing internal actions (Wang et al., 2024b). These agent frameworks typically consist of several components, including reasoning, tool usage, memory, and multi-agent debate (Masterman et al., 2024), many of which can be seamlessly integrated into fact-checking pipelines to enhance the performance of traditional fact-checking systems. For example, recent works have endowed systems with the ability to call external tools, such as search engines (Chern et al., 2023; Wang et al., 2023; Wei et al., 2024; Cheng et al., 2024), recognizing that many claims in the field require additional information for verification. During the verification stage, Sun et al. (2024) proposed a Markov Chain-based multi-agent debate approach to ensure more rigorous verification by enabling collaborative decision-making among agents based on retrieved evidence. Our work differs from previous approaches by combining the evidence retrieval and verification stages, leveraging agents' reasoning and tool-use capabilities to more closely simulate human cognitive processes in fact-checking.

3 Framework

Assessing the factual accuracy of long-form text presents significant challenges (Min et al., 2023). To address this, prior approaches have broken down the text into individual checkworthy claims (Chern et al., 2023). These sentences, referred to as atomic claims, are fact-checked individually, with their factuality scores aggregated to evaluate the overall factual accuracy of the original text. Previous research indicates that verifying the factuality of atomic claims is the most challenging step in this process (Wang et al., 2023). **Our work therefore focuses on this critical task: determining the factual accuracy of individual atomic claims, classifying each as either *True* or *False*.**

3.1 FIRE

We present FIRE, a simple yet effective agent-based framework for interactive claim verification through web searches. As illustrated in Figure 1, FIRE takes an atomic claim as input and outputs a binary label indicating whether the claim is factual or non-factual. The framework consists of three key components: *Final Answer or Next Search Query*, *Web Search*, and *Final Verification*, each of which we will explain below.

Final Answer or Next Search Query We introduce a unified method, *Final Answer or Next Search Query* $f(\cdot)$, which integrates claim verification with search query generation. This component processes an atomic claim c and determines whether to produce a final answer a or to generate an additional search query q. This decision is informed by both an external evidence set E, derived from search results, and the internal knowledge k of the language model, acquired during pre-training. To facilitate this, we incorporate confidence estimation into the reasoning process. The model outputs a final answer a when its confidence is sufficiently high; otherwise, it generates an additional query q. This process can be formalized as follows:

$$f(c, E, k) = \begin{cases} a, & \text{if confident} \\ q, & \text{if not confident} \end{cases}$$

This method offers greater flexibility by eliminating the need to retrieve a fixed number of evidence items before verification, thereby largely reducing search costs. A detailed description of the prompt used for this component is provided in Appendix A.

Web search When the language model determines that a web search is necessary by issuing a search query q, we utilize Google Search through the SerpAPI[1]. This service returns the search re-

[1] https://serpapi.com

sults as new evidence e, which we then append to the existing evidence set E to form the new evidence set E' for the next iteration.

$$E' = E \cup e, e = \text{Search}(q)$$

Final Verification Due to the inherent difficulty in confidently verifying certain claims, even after retrieving supplementary evidence, we establish an upper limit on the number of retrieval steps. Upon reaching this limit, the model is required to conduct a final verification $f'(\cdot)$ based on all previously retrieved evidence. The detailed prompt for this process is provided in Appendix B.

$$\begin{cases} a = f'(c, E, k), & n \geq N \\ e = \text{Search}(q), & n < N \end{cases}$$

3.2 Prevention of Repetitive Search Queries

In our preliminary studies, we identified a recurring issue with sequential search query generation using language models: the tendency of these models to generate repetitive queries. This occurs even when the models are explicitly instructed to generate queries targeting new, claim-relevant information. As a result, identical queries are repeatedly submitted to web search tools, leading to inefficient use of search resources. To address this issue, we investigate following methods for enhancing search query generation and reducing repetition.

Early Termination The iterative process is terminated when consecutive queries or retrieved results exhibit a high degree of similarity, indicating diminishing returns.

Diversity Prompt We introduce additional prompts to encourage the model to generate more diverse queries when consecutive similar queries or search results are detected.

3.3 Prevention of Verification Overconfidence

LLMs can exhibit strong calibration abilities across diverse tasks (Kadavath et al., 2022; Geng et al., 2024). Consequently, they are aware of their confidence levels during the claim verification process. However, our preliminary analysis reveals that LLMs often demonstrate excessive strictness and unwarranted confidence in certain cases, leading to errors. Considering this, we explore several techniques to prevent overconfidence in verification:

At Least One/Two At Least One requires models to retrieve at least one evidence during the verification, which increase the probability of eliminating overconfidence. Similarly, we also adopted a more aggressive approach At Least Two to retrieve a second evidence to reduce the uncertainty.

Inclusive Prompt In this setting, we prompt models to be "less strict, open-minded and avoid being over confident" to encourage models to reflect on their confidence level of answers.

4 Experiments Setup

4.1 Datasets

In our study, we utilized four datasets from prior research that align with our experimental setup: FacTool (Chern et al., 2023), FELM (Chen et al., 2023), Factcheck-Bench (Wang et al., 2023), and BingCheck (Li et al., 2024b).

FacTool and FELM provide factuality claims across multiple domains. From these, we selected instances requiring world knowledge for verification, which we refer to as FacTool-QA and FELM-WK, both with binary annotations (*True* or *False*). Factcheck-Bench and BingCheck, however, pose a four-label classification task: *supported*, *partially supported*, *not supported*, and *refuted*. Following Iqbal et al. (2024), we merged *supported* and *partially supported* into *True*, treated *refuted* as *False*, and excluded *not supported*. Given BingCheck's imbalance (3,581 *True*, 42 *False*), we sampled 100 *True* claims for our test set. For FELM-WK, we used un-split claims to avoid loss of context. Dataset statistics are in Table 2.

In our experiments, we first use the Factcheck-Bench dataset as a development set to optimize the settings for our framework. We then evaluate its performance on the remaining three datasets, comparing it with other competitive fact-checking systems.

4.2 Language Models

We investigate several state-of-the-art (SOTA) language models, including proprietary models from two prominent families: GPT models (OpenAI, 2024a,b) and Claude models (Anthropic, 2024), as detailed in Table 1. In addition, we assess two open-source models: LLaMA 3.1-Inst 8B (Dubey et al., 2024) and Mistral-Inst 7B (Jiang et al., 2023).

Family	Name
GPT	GPT-4o, GPT-4o-mini, o1-preview, o1-mini
Claude	Claude-3 Haiku, Claude-3 Opus, Claude-3.5 Sonnet
LLaMA	LLaMA 3.1-Inst 8B
Mistral	Mistral-Inst 7B

Table 1: **Model families and specific model names used in this study.**

Dataset	#True	#False	Total
Factcheck-Bench	472	159	631
FacTool-QA	177	56	233
FELM-WK	99	85	184
BingCheck	100	42	142

Table 2: **Statistics of the datasets after processing.**

4.3 Compared Fact-checking Frameworks

We select several SOTA fact-checking frameworks for comparison. Additionally, we introduce two baseline models: Random and Always True/False. To further assess the impact of LLM reasoning in fact-checking, we include an ablation setting.

FACTOOL is adaptable across domains and tasks, using a tool-augmented framework that integrates external tools like Google Search and Python interpreters to assess the factuality of content from large language models. However, this can introduce complexity and depend on the accuracy of these external tools.

FACTCHECK-GPT excels in fine-grained factuality evaluation through a detailed benchmark with annotations at the claim, sentence, and document levels. While resource-intensive, it provides valuable insights into specific stages of factual inaccuracies.

SAFE uses a search-augmented approach to verify long-form content by breaking it down into individual facts and checking them via Google Search. This method is cost-effective compared to human annotation but depends on the reliability of search engine results, which can vary and introduce biases.

Random assigns the predicted label for each claim in the test set randomly, choosing between *True* and *False* with equal probability.

Always True/False is an approach that always predicts a single label – either *True* or *False* – for all claims in the test set.

FIRE (No Reason) utilizes the same framework as FIRE; however, it is explicitly instructed not to articulate its reasoning process in the output. This modification aims to assess the impact of explicitly stating the step-by-step reasoning process on the results.

4.4 Evaluation Metrics

In this work, we investigate the trade-off between computational cost and fact-checking performance.

Performance We evaluate precision, recall, and F1 scores for both positive and negative classes.

Computational Cost We report the financial costs of LLM API calls for proprietary models and GPU rental expenses for open-source models, alongside an analysis of API costs from search engine queries and a breakdown of the total time spent on the fact-checking process. The experiments using open-source models were conducted on an NVIDIA RTX 6000 GPU at an estimated cost of \$0.79 per hour, while search queries via SerpAPI incurred approximately \$0.00105 per search.

5 Results

In this section, we first present preliminary studies on Factcheck-Bench (§ 5.1), focusing on three key aspects: language models, prevention of repetitive search queries, and prevention of verification overconfidence. These studies aim to identify the most appropriate configurations for our framework. Subsequently, we compare FIRE to other strong fact-checking frameworks across three additional datasets (§ 5.2) to evaluate the generalization capabilities of our approach.

5.1 Preliminary studies

Language Models We present a performance comparison of various language models in Table 3. Overall, proprietary language models generally outperform open-source models, likely due to their larger size and more sophisticated training in reasoning and tool utilization. Among the proprietary models, the latest and most advanced offerings from different organizations—specifically o1-preview from OpenAI and Claude-3.5 Sonnet from Anthropic—exhibit the best performance. Although the more economical model, GPT-4o-mini, performs slightly worse than the top-performing o1-preview, it offers a cost savings of 766 times.

LLM	LLM+Search Cost ($)	Label = True			Label = False		
		Prec	Recall	F1	Prec	Recall	F1
GPT-4o-mini	**0.19+0.44**	0.91	0.84	0.87	0.61	0.74	0.67
GPT-4o	10.45+1.47	0.92	0.79	0.85	0.56	0.79	0.66
o1-preview	145.66+0.80	0.91	**0.86**	**0.88**	**0.64**	0.75	0.69
o1-mini	20.06+1.13	0.89	0.81	0.85	0.56	0.71	0.62
Claude-3 Haiku	0.56+0.85	0.9	0.81	0.85	0.56	0.73	0.64
Claude-3 Opus	48.64+1.43	0.92	0.81	0.86	0.58	0.79	0.67
Claude-3.5 Sonnet	13.21+1.63	**0.94**	0.79	0.86	0.58	**0.85**	**0.69**
LLaMA 3.1-Inst 8B	3.95+2.27	0.89	0.74	0.8	0.48	0.72	0.57
Mistral-Inst 7B	1.84+1.22	0.85	0.67	0.75	0.4	0.66	0.5

Table 3: **Fact-checking performance and cost comparisons between different language models within FIRE on Factcheck-Bench.**

Window Size	Diversity Prompt	LLM+Search Cost ($)	Label = True			Label = False		
			Prec	Recall	F1	Prec	Recall	F1
2	✗	0.17+0.29	**0.92**	0.83	**0.87**	**0.61**	**0.77**	**0.68**
	✓	0.16+0.29	0.91	0.81	0.86	0.57	0.76	0.65
3	✗	0.17+0.36	0.91	0.82	**0.87**	0.60	0.77	0.67
	✓	0.18+0.36	0.91	0.82	0.86	0.59	0.76	0.66
4	✗	0.18+0.39	0.91	0.81	0.86	0.57	0.76	0.65
	✓	0.18+0.39	0.91	0.82	0.86	0.59	0.76	0.66
Default	-	0.19+0.44	0.91	**0.84**	**0.87**	**0.61**	0.74	0.67

Table 4: **FIRE performance across various window sizes, with and without the use of prompts for generating diverse queries on Factcheck-Bench.**

This suggests that for fact-checking tasks, the most advanced models may not be necessary; GPT-4o-mini can serve as a sufficiently capable alternative at a significantly lower cost. We will continue our preliminary studies using GPT-4o-mini.

Prevention of Repetitive Search Queries We conducted an experimental analysis to evaluate the impact of **Early Termination** and **Diversity Prompt** on mitigating the generation of repetitive search queries. To assess query similarity, we employed Sentence-BERT (all-MiniLM-L6-v2; Reimers and Gurevych (2019)) with a similarity threshold of 0.9, as established by Shashavali et al. (2019). Table 4 presents experimental results, where **window size** refers to the predefined number of consecutive similar queries or retrieval results. Once this threshold is reached, early termination is triggered to prevent further query generation and retrieval. If the model generates queries or retrieves results exhibiting high similarity within this window, the system also activates an early stopping mechanism. The results indicate that optimizing the similarity window size effectively reduces search costs without compromising the model's performance. However, our findings suggest that the diversity prompt does not enhance performance. In our optimal configuration, we selected a window size of 2 without utilizing the diversity prompt.

Approach	LLM+Search Cost ($)	Label = True			Label = False		
		Prec	Recall	F1	Prec	Recall	F1
At Least One	0.21+0.83	**0.92**	0.81	0.86	0.58	**0.78**	0.67
At Least Two	0.22+0.87	0.91	0.79	0.84	0.55	0.77	0.64
Inclusive	0.20+**0.42**	0.91	0.81	0.86	0.58	0.77	0.66
Default	**0.19**+0.44	0.91	**0.84**	**0.87**	**0.61**	0.74	**0.67**

Table 5: **FIRE performance using different verification overconfidence prevention approaches on Factcheck-Bench.**

Prevention of Verification Overconfidence We present the performance and cost of various overconfidence prevention approaches for verification on Factcheck-Bench in Table 5. Interestingly, the **At Least One/Two** settings, which aggressively retrieve additional evidence, result in higher search costs without improving fact-checking performance compared to the **Default** setting, where no explicit constraints are placed on web search. This supports our hypothesis that most atomic claims are relatively straightforward and do not require extensive external web searches for verification. In fact, introducing additional searches may introduce noise, negatively impacting performance. The **Inclusive** setting encourages models to be more flexible and open to alternative interpretations of evidence, which reduces the need for queries but also leads to lower overall performance. Based on these observations, we maintain the **Default** setting, leveraging the language model's reasoning capabilities without imposing additional search constraints.

5.2 Comparisons to Other Frameworks

We present a performance comparison of our framework against other frameworks in Table 6 and a cost analysis in Table 7. As shown, all frameworks exhibit similar performance, with a small gap of approximately 0.2 to 0.5. Our framework, using GPT-4o, performs slightly better, achieving superior results on 7 out of 18 metrics, followed closely by SAFE with GPT-4o at 6 metrics. This suggests that all frameworks can effectively perform fact-checking for most claims, although they may encounter difficulties with challenging examples, which we analyze further in § 6. In terms of model size, GPT-4o generally outperforms GPT-4o-mini across most frameworks, indicating that larger models are more effective in detecting misinformation. However, the performance improvement is modest, as it comes with an average increase of 16.7 times in LLM costs and an additional 3-fold in-

Framework	LLM	FacTool-QA						FELM-WK						BingCheck					
		Label = True			Label = False			Label = True			Label = False			Label = True			Label = False		
		Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
Random	-	0.81	0.47	0.59	0.28	0.64	0.39	0.75	0.49	0.59	0.30	0.57	0.39	0.77	0.67	0.72	0.40	0.52	0.45
Always True	-	0.76	1.0	0.86	0	0	0	0.72	1.0	0.84	0	0	0	0.70	1.0	0.83	0	0	0
Always False	-	0	0	0	0.24	1.0	0.39	0	0	0	0.28	1.0	0.44	0	0	0	0.30	1.0	0.46
FACTOOL	GPT-4o	0.88	0.81	0.84	0.52	0.66	0.58	0.69	0.53	0.60	0.57	0.73	0.64	0.86	0.57	0.68	0.43	0.79	0.56
	GPT-4o-mini	**0.92**	0.68	0.78	0.45	**0.82**	0.58	0.67	0.37	0.48	0.51	**0.78**	0.62	**0.92**	0.55	0.69	0.45	**0.88**	0.60
FACTCHECK-GPT	GPT-4o	0.90	0.79	0.84	0.52	0.71	0.60	0.67	0.68	0.67	0.61	0.61	0.61	0.85	0.70	0.77	0.50	0.71	0.59
	GPT-4o-mini	0.85	0.80	0.82	0.47	0.56	0.51	0.61	0.50	0.55	0.51	0.62	0.56	0.88	0.78	0.83	0.60	0.76	0.67
SAFE	GPT-4o	**0.92**	**0.88**	**0.90**	**0.66**	0.77	0.71	**0.70**	0.80	0.75	0.72	0.60	0.65	0.84	0.90	0.87	**0.71**	0.60	0.65
	GPT-4o-mini	**0.92**	0.82	0.87	0.58	0.79	0.67	0.61	0.76	0.68	0.61	0.44	0.51	0.86	0.81	0.84	0.60	0.69	0.64
FIRE	GPT-4o	**0.92**	**0.88**	**0.90**	0.65	0.71	0.68	**0.70**	**0.86**	**0.77**	**0.77**	0.54	0.63	0.86	0.88	0.87	0.70	0.67	0.68
	GPT-4o-mini	0.87	**0.88**	0.87	0.60	0.59	0.59	0.63	0.82	0.71	0.67	0.44	0.53	0.87	**0.91**	**0.88**	**0.74**	0.67	**0.70**
FIRE (No Reason)	GPT-4o	0.88	0.86	0.87	0.60	0.64	0.62	**0.70**	0.85	**0.77**	**0.77**	0.58	**0.66**	0.85	0.89	0.87	0.70	0.62	0.66
	GPT-4o-mini	0.87	0.84	0.86	0.55	0.61	0.58	0.65	0.84	0.73	0.71	0.47	0.57	0.84	0.87	0.85	0.66	0.6	0.62

Table 6: **Performance comparisons between different frameworks across multiple datasets.**

Framework	LLM	LLM Cost ($)	Search Cost ($)	Time (hrs)
FACTOOL	GPT-4o	24.76	3.67	2.92
	GPT-4o-mini	1.49	3.67	2.34
FACTCHECK-GPT	GPT-4o	21.41	-	4.25
	GPT-4o-mini	1.28	-	4.09
SAFE	GPT-4o	6.34	2.93	4.62
	GPT-4o-mini	0.43	2.93	4.25
FIRE	GPT-4o	3.35	0.60	1.31
	GPT-4o-mini	0.14	**0.20**	1.25
FIRE (No Reason)	GPT-4o	1.65	0.68	0.57
	GPT-4o-mini	**0.07**	0.59	**0.54**

Table 7: **Time and USD cost for evaluating the total 559 atomic claims in FacTool-QA, FELM-WK, and BingCheck.** We use SerperAPI for FACTOOL, SAFE and FIRE for search, while FACTCHECK-GPT has its own implemented scrapping technique.

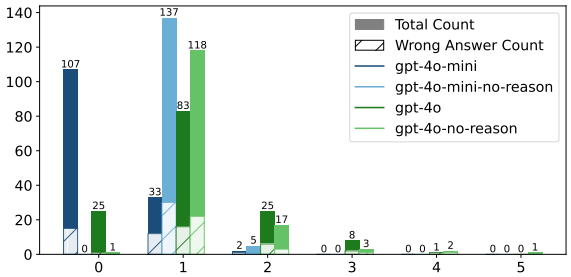

Figure 2: **The effect of reasoning on the number of searches using GPT-4o and GPT-4o-mini within FIRE on BingCheck.** The x-axis shows the number of web searches, while the y-axis denotes the number of instances.

crease in search costs when using FIRE. When considering all frameworks with GPT-4o-mini, FIRE demonstrates cost savings of 7.6 times in LLM expenses and 16.5 times in search costs compared to other frameworks. Therefore, we argue that FIRE, when paired with GPT-4o-mini, presents a compelling option for the large-scale production of fact-checking systems.

Figure 2 illustrates the impact of reasoning on the number of web searches conducted by GPT-4o and GPT-4o-mini when using BingCheck. Notably, GPT-4o-mini demonstrates a high level of confidence in making verifications when it is allowed to articulate its reasoning process, resulting in the majority of judgments being made without any searches. Conversely, when not permitted to express its reasoning, there is a significant decrease in the number of instances with zero searches; most cases now involve at least one search, indicating a marked reduction in GPT-4o-mini's confidence in its judgments. This observation aligns with previous findings that the presence of CoT reasoning correlates with increased confidence in the model's answers (Wang and Zhou, 2024). While GPT-4o also shows a decline in confidence when it is not allowed to search, the decrease is less pronounced than that observed in GPT-4o-mini. By combining the performance and cost results presented in Table 6 and Table 7, we find that, in the absence of a reasoning process, the costs associated with LLMs can be reduced through fewer completion tokens. However, this reduction leads to increased search costs, resulting in overall performance that is inferior to scenarios in which the models are permitted to engage in step-by-step reasoning. Furthermore, the step-by-step reasoning approach facilitates more effective error analysis.

6 Error Analysis

To identify weaknesses in our fact-checking system, we manually examine failed cases of three datasets: FELM-WK, FacTool-QA, and BingCheck, analyzing whether the majority of failures is attributed to inadequate retrieved evidence or to flaws in the LLM verification process, despite the availability of reliable evidence.

Major Issue	Error Type Description	FELM	FacTool	BingCheck	#Total
I. Dataset Issue	**1. Not a claim**, e.g. a claim only has a name *Elvis Presley.*	12	1	0	13
	2. Unclear, ambiguous or subjective claim e.g. there is no record of how many sons **he** had.	11	7	2	20
	3. False gold labels, i.e., the original annotated label might be wrong. For example, claim *choosing organic and local foods that are in season can reduce emissions from transporting food from far away* is labeled as false	10	0	1	11
II. Knowledge Issue	**4. Complicated science domain expert knowledge** is needed to judge, like astronomy.	3	0	1	4
	5. Inaccurate parametric knowledge. LLM-based verifiers make wrong verification due to the incorrect parametric knowledge stored in LLMs.	3	6	7	16
	6. Insufficient or inaccurate externally collected knowledge (evidence), involving three scenarios: (i) no external evidence, model makes wrong reasoning by itself; (ii) collected evidence is insufficient to cover all aspects mentioned by a long claim; (iii) collected evidence is inaccurate (e.g., evidence contain statement *More than 430 species of mammal are found in the Amazon* when the correct number is 427).	9	15	6	**30**
III. LLM Reasoning	**7. Incorrect reasoning**, e.g., the claim mentioned A while the model dismissed in the reasoning process, or the claim did not mention A while the model hallucinated A	6	8	4	18
	8. Strict reasoning includes two situations: (i) strictly depending on the collect evidence to make decision leads to wrong verification, while if it combines commonsense and collected evidence to analyze, it can verify correctly; (2) strict reasoning based on parametric knowledge. E.g. regarding *Word-Cross Puzzle* and *FUN Word-Cross Puzzle* are different.	5	0	3	8
IV. Debatable Opinion	**9. Debatable Opinions** on controversial topics, e.g. actual origins are debated for claim *Fortune cookies made their way to San Francisco in the late 1800s and early 1900s through Japanese immigrants.*	7	8	0	15
Total		**66**	**45**	**24**	**135**

Table 8: **Datasets Error distribution, grouped into nine fine-grained types under four major issues.**

We summarized errors into four major issues and nine error types. Among the total number of 135 failed claims, there are 44 cases falling into challenges of (I) inaccurate identification of check-worthy claims and false gold labels in the original datasets, 50 claims are due to (II) inaccurate or insufficient knowledge applied to verification, either internally extracted from LLM parameters or externally collected from web pages. The rest 26 and 15 cases result from LLM reasoning ability and debatable opinions over some topics, respectively, as shown in Table 8.

The major issue lies in collecting sufficient evidence, especially for long claims containing many aspects to verify. This can be approached by decomposing "atomic claims" from the original dataset into the real granularity of "atomic", each containing only 1-3 pieces of information. The second problem focus on the quality of benchmarking datasets, particularly FELM-WK that includes many ungrounded claims and labels (Li et al., 2024a), which may lead to ineffective comparisons between fact-checking systems. Interestingly, beyond incorrect reasoning, overly-strict reasoning by exact matching between the claim and collected evidence can also lead to verification errors. For example, LLMs label a claim as false when the claim states *FUN Word-Cross Puzzle* while evidence mentions *Word-Cross Puzzle*. Additionally, some claims can be viewed as true from one perspective but false from another, as seen in debates over the origins of fortune cookies, where the truth of related claims is debatable.

Considering above, to further advance the field of fact-checking, we highlight the need for improved benchmarking datasets, a stronger focus on verifying fine-grained claims, and strategies to guide LLMs in performing verification under more flexible reasoning conditions, such as semantic alignment, rather than relying exclusively on exact matches.

7 Conclusions and Future Work

Conventional fact-checking systems typically separate the steps of evidence retrieval and claim verification, leading to suboptimal utilization of the verification models' internal knowledge. To address this, we propose FIRE, a novel framework that integrates evidence retrieval and claim verification in an iterative process. FIRE enables LLMs to leverage their internal knowledge for judgment and only rely on external evidence retrieval when uncertain. Our experiments on multiple datasets demonstrate that FIRE not only slightly improves accuracy but also reduces LLM computation costs by an average of 7.6 times and search costs by 16.5 times, making it highly efficient for production use. Additionally, we performed a detailed error analysis, which revealed issues with the benchmarking datasets quality. These findings highlight the need for further research into edge cases, rather than relying solely on automatic metrics for evaluation.

We identify several promising directions for future work: (1) Integrating memory banks to store verification results, enabling the system to leverage previous results instead of repeatedly executing the entire process; (2) Expanding the system to support additional modalities, such as code and images.

Limitations

We acknowledge certain limitations in this work that we plan to address in future research. First, to maintain the efficiency of our framework, we implement the *Final Answer or Next Search Query* in a compact manner to retrieve evidence, assess confidence on knowledge, and verify the final answer within a single step. Ideally, this can be implemented via separate process which involves a standalone confidence estimation step. This would offer greater flexibility and interpretability. We plan to leave this to our future work. Second, for fair comparison across multiple fact-checking datsets, our system adopts a binary labeling scheme (*True* or *False*) and standarlizes the labels across datasets. However, this may not fully reflect the complexity of factual labels in the real world. We intend to incorporate fine-grained labeling schemes in the future.

Ethical Statement and Broad Impact

Data License A primary ethical consideration is the data license. We reused pre-existing dataset, FactBench, FACTOOL, FELM-WK, BingCheck, which have been publicly released and approved for research purposes. We adhere to the intended usage of all these dataset licenses.

Ethical Statement We acknowledge that our system relies on LLMs, which can sometimes produce biased or incorrect judgments due to the data used in their pre-training or biases present in external sources. Additionally, there is the risk of over-reliance on the system for making critical factual judgments without human oversight. To mitigate these risks, we strongly encourage human reviewers to be involved in decision-making, especially in high-stakes domains such as legal, political, or medical contexts.

Broad Impact FIRE has the potential to advance the field of automated fact-checking by enhancing its efficiency and accessibility. Its capability to iteratively retrieve evidence while minimizing computational costs will empower a broader range of users—including journalists, researchers, and the general public—to verify factual information with greater ease. Furthermore, FIRE can be applied to large-scale implementations, such as integration into search engines and social media platforms, thereby contributing to efforts to combat the spread of misinformation.

Acknowledgments

References

Mubashara Akhtar, Michael Schlichtkrull, Zhijiang Guo, Oana Cocarascu, Elena Simperl, and Andreas Vlachos. 2023. Multimodal automated fact-checking: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5430–5448. Association for Computational Linguistics.

Anthropic. 2024. Introducing the next generation of claude.

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

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: benchmarking factuality evaluation of large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, Hongzhi Zhang, Fuzheng Zhang, Di Zhang, Kun Gai, and Ji-Rong Wen. 2024. Small agent can also rock! empowering small language models as hallucination detector. *CoRR*, abs/2406.11277.

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. *CoRR*, abs/2307.13528.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large

language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6577–6595. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. 2023. AVIS: autonomous visual information seeking with large language model agent. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2024. Openfactcheck: A unified framework for factuality evaluation of llms. *CoRR*, abs/2408.11832.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Haonan Li, Xudong Han, Hao Wang, Yuxia Wang, Minghan Wang, Rui Xing, Yilin Geng, Zenan Zhai, Preslav Nakov, and Timothy Baldwin. 2024a. Loki: An open-source tool for fact verification. *Preprint*, arXiv:2410.01794.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu Zhang. 2024b. Self-checker: Plug-and-play modules for fact-checking with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 163–181. Association for Computational Linguistics.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The landscape of emerging AI agent architectures for reasoning, planning, and tool calling: A survey. *CoRR*, abs/2404.11584.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024a. Hello gpt-4o.

OpenAI. 2024b. Introducing openai o1-preview.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

D. Shashavali, V. Vishwjeet, Rahul Kumar, Gaurav Mathur, Nikhil Nihal, Siddhartha Mukherjee, and Suresh Venkanagouda Patil. 2019. Sentence similarity techniques for short vs variable length text using word embeddings. *Computación y Sistemas*, 23(3).

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2024. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. *CoRR*, abs/2406.03075.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *CoRR*, abs/2402.10200.

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. 2023. Factcheck-gpt: End-to-end fine-grained document-level fact-checking and correction of LLM output. *CoRR*, abs/2311.09000.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024a. Factuality of large language models in the year 2024. *CoRR*, abs/2402.02420.

Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024b. What are tools anyway? A survey from the language model perspective. *CoRR*, abs/2403.15452.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *CoRR*, abs/2403.18802.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023a. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351, Prague, Czechia. Association for Computational Linguistics.

Zhuohan Xie, Miao Li, Trevor Cohn, and Jey Han Lau. 2023b. Deltascore: Fine-grained story evaluation with perturbations. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5317–5331. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

A Prompts for Verification

Default prompt We use the following prompt to guide the language model in verifying the atomic claim, determining whether to provide a final judgment or issue an additional Google search query based on the current status. The prompt will output reason or explanation for the verification process.

```
_FINAL_ANSWER_OR_NEXT_SEARCH_FORMAT = f"""
    Instructions:
1. You are provided with a STATEMENT and
    relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
    accuracy of the STATEMENT.
3. Before presenting your conclusion, think
    through the process step-by-step. Include
    a summary of the key points from the
    KNOWLEDGE as part of your reasoning.
4. If the KNOWLEDGE allows you to confidently
    make a decision, output the final answer
    as a JSON object in the following format:
  {{
    "final_answer": "{_Factual_LABEL}" or "{
        _Non_Factual_LABEL}"
  }}
5. If the KNOWLEDGE is insufficient to make a
    judgment, issue ONE Google Search query
    that could provide additional evidence.
    Output the search query in JSON format, as
     follows:
  {{
    "search_query": "Your Google search query
        here"
  }}
6. The query should aim to obtain new
    information not already present in the
    KNOWLEDGE, specifically helpful for
    verifying the STATEMENT's accuracy.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

No Reason prompt To improve efficiency, we opted for this setting to output only the label.

```
_FINAL_ANSWER_OR_NEXT_SEARCH_FORMAT = f"""
    Instructions:
1. You are provided with a STATEMENT and
    relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
    accuracy of the STATEMENT.
3. Before presenting your conclusion, think
    through the process step-by-step. Include
    a summary of the key points from the
    KNOWLEDGE as part of your reasoning.
4. If the KNOWLEDGE allows you to confidently
    make a decision, output the final answer
    as a JSON object in the following format:
  {{
    "final_answer": "{_Factual_LABEL}" or "{
        _Non_Factual_LABEL}"
  }}
```

```
5. If the KNOWLEDGE is insufficient to make a
     judgment, issue ONE Google Search query
     that could provide additional evidence.
     Output the search query in JSON format, as
      follows:
   {{
     "search_query": "Your Google search query
         here"
   }}
6. The query should aim to obtain new
     information not already present in the
     KNOWLEDGE, specifically helpful for
     verifying the STATEMENT's accuracy.
7. Do not provide any additional information
     or reasoning in the output. Only output
     the JSON object.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

At Least One prompt At Least One prompt requires models to retrieve at least one evidence during the verification.

```
_FINAL_ANSWER_OR_NEXT_SEARCH_FORMAT = f"""
     Instructions:
1. You are provided with a STATEMENT and
     relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
     accuracy of the STATEMENT.
3. Before presenting your conclusion, think
     through the process step-by-step. Include
     a summary of the key points from the
     KNOWLEDGE as part of your reasoning.
4. If the KNOWLEDGE allows you to confidently
     make a decision, output the final answer
     as a JSON object in the following format:
   {{
     "final_answer": "{_Factual_LABEL}" or "{
         _Non_Factual_LABEL}"
   }}
5. If the KNOWLEDGE is insufficient to make a
     judgment, issue ONE Google Search query
     that could provide additional evidence.
     Output the search query in JSON format, as
      follows:
   {{
     "search_query": "Your Google search query
         here"
   }}
6. The query should aim to obtain new
     information not already present in the
     KNOWLEDGE, specifically helpful for
     verifying the STATEMENT's accuracy.
7. If the KNOWLEDGE is empty, please issue ONE
      Google Search query immediately.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

At Least Two prompt At Least Two prompt is a more aggressive approach to retrieve minimum two evidence before verification.

```
_FINAL_ANSWER_OR_NEXT_SEARCH_FORMAT = f"""
     Instructions:
1. You are provided with a STATEMENT and
     relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
     accuracy of the STATEMENT.
3. Before presenting your conclusion, think
     through the process step-by-step. Include
     a summary of the key points from the
     KNOWLEDGE as part of your reasoning.
4. If the KNOWLEDGE allows you to confidently
     make a decision, output the final answer
     as a JSON object in the following format:
   {{
     "final_answer": "{_Factual_LABEL}" or "{
         _Non_Factual_LABEL}"
   }}
5. If the KNOWLEDGE is insufficient to make a
     judgment, issue ONE Google Search query
     that could provide additional evidence.
     Output the search query in JSON format, as
      follows:
   {{
     "search_query": "Your Google search query
         here"
   }}
6. The query should aim to obtain new
     information not already present in the
     KNOWLEDGE, specifically helpful for
     verifying the STATEMENT's accuracy.
7. If the KNOWLEDGE is empty or there is only
     ONE evidence in the KNOWLEDGE, please
     issue ONE Google Search query immediately.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

Inclusive In this setting, we prompt models to be "less strict, open-minded and avoid being over confident" to encourage models to reflect on their confidence level of answers.

```
_FINAL_ANSWER_OR_NEXT_SEARCH_FORMAT = f"""
     Instructions:
1. You are provided with a STATEMENT and
     relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
     accuracy of the STATEMENT.
3. Before presenting your conclusion, think
     through the process step-by-step. Include
     a summary of the key points from the
     KNOWLEDGE as part of your reasoning.
4. If the KNOWLEDGE allows you to confidently
     make a decision, output the final answer
     as a JSON object in the following format:
   {{
     "final_answer": "{_Factual_LABEL}" or "{
         _Non_Factual_LABEL}"
   }}
```

```
5. If the KNOWLEDGE is insufficient to make a
     judgment, issue ONE Google Search query
     that could provide additional evidence.
     Output the search query in JSON format, as
      follows:
   {{
     "search_query": "Your Google search query
         here"
   }}
6. The query should aim to obtain new
     information not already present in the
     KNOWLEDGE, specifically helpful for
     verifying the STATEMENT's accuracy.
7. Please be more open-minded and less strict
     in your evaluation. Avoid being overly
     confident, and consider the possibility of
      alternative interpretations or
     uncertainties in the evidence.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

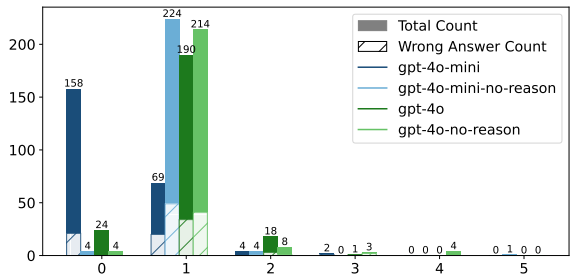

Figure 3: **The effect of reasoning on the number of searches using GPT-4o and GPT-4o-mini within FIRE on FacTool-QA.**

B Prompt for Final Verification

Upon reaching the maximum number of steps, we issue the following prompt to compel the language model to make a final judgment based on the accumulated information.

```
_MUST_HAVE_FINAL_ANSWER_FORMAT = f"""
     Instructions:
1. You are provided with a STATEMENT and
     relevant KNOWLEDGE points.
2. Based on the KNOWLEDGE, assess the factual
     accuracy of the STATEMENT.
3. Before presenting your final answer, think
     step-by-step and show your reasoning.
     Include a summary of the key points from
     the KNOWLEDGE as part of your reasoning.
4. Your final answer should be either "{
     _Factual_LABEL}" or "{_Non_Factual_LABEL
     }".
5. Format your final answer as a JSON object
     in the following structure:
   {{
     "final_answer": "{_Factual_LABEL}" or "{
         _Non_Factual_LABEL}"
   }}
6. Do not include any other information or
     reasoning in the output. Only provide the
     JSON object.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
"""
```

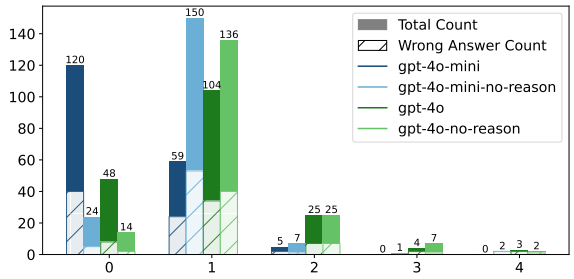

Figure 4: **The effect of reasoning on the number of searches using GPT-4o and GPT-4o-mini within FIRE on FELM-WK.**