EMULATE: A Multi-Agent Framework for Determining the Veracity of Atomic Claims by Emulating Human Actions

Spencer Hong   Meng Luo   Xinyi Wan
National University of Singapore
{spencer.hong, mluo, wan.xinyi}@u.nus.edu

Abstract

Determining the veracity of atomic claims is an imperative component of many recently proposed fact-checking systems. Many approaches tackle this problem by first retrieving evidence by querying a search engine and then performing classification by providing the evidence set and atomic claim to a large language model, but this process deviates from what a human would do in order to perform the task. Recent work attempted to address this issue by proposing iterative evidence retrieval, allowing for evidence to be collected several times and only when necessary. Continuing along this line of research, we propose a novel claim verification system, called EMULATE, which is designed to better emulate human actions through the use of a multi-agent framework where each agent performs a small part of the larger task, such as ranking search results according to predefined criteria or evaluating webpage content. Extensive experiments on several benchmarks show clear improvements over prior work, demonstrating the efficacy of our new multi-agent framework. Our code is available at `https://github.com/qqqube/EMULATE`.

1 Introduction

To prevent the spread of misinformation, a multitude of automated fact-checking systems have recently been proposed in the natural language processing community (Xie et al., 2025; Wang et al., 2024; Singal et al., 2024; Kim et al., 2024; Chen et al., 2024; Chern et al., 2023; Pan et al., 2023; Wang and Shu, 2023). For example, Chern et al. (2023) and Wang et al. (2024) introduced frameworks that break texts into atomic claims and equip LLMs with the ability to use web search tools to retrieve evidence for verifying the claims. Other works have also considered taking iterative approaches for search query generation (Wei et al., 2024) as well as the whole retrieval/verification process (Xie et al., 2025).

Continuing along this line of research, we propose a novel system, called EMULATE, that takes an atomic claim as input and determines the veracity of the claim by retrieving evidence from the web and mimicking human actions. More specifically, we employ a multi-agent framework that consists of agents for generating search queries, determining the credibility and relevance of search results, evaluating webpage content, assessing collections of evidence, and performing classification. By having each agent execute a small part of the larger task, our system can successfully guide the underlying language models in retrieving important information from external resources which ultimately leads to an amelioration in classification performance.

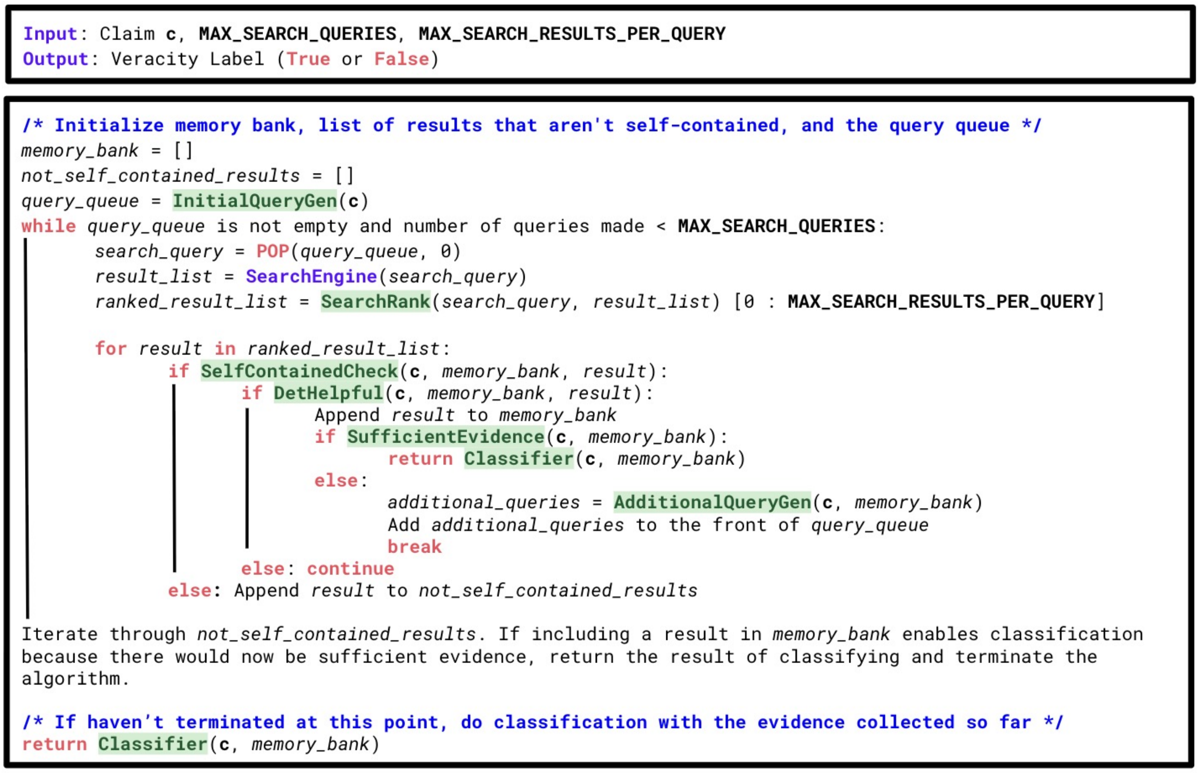
The closest work to ours is FIRE (Xie et al., 2025), which consists of three components: one for either outputting the final answer or generating the next search query, another for making web searches and retrieving the snippets of the search results, and a third for final verification after a maximum number of retrieval steps has been reached. Though FIRE also makes use of several agents, our framework further breaks down the fact verification process by trying to understand why additional evidence is needed at each step, which enables the system to ignore redundant and irrelevant search results and generate high-quality queries that can better enhance the system's evidence set.

To evaluate the efficacy of our framework, we perform experiments on a variety of fact-checking benchmarks. Our results show a clear improvement over prior work and demonstrate the effectiveness of using multi-agent systems to tackle complex tasks like fact verification.

2 Related Work

Many existing automatic fact-checking pipelines adopt the *Decompose-Then-Verify* paradigm, which

Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER), pages 179–183
July 31, 2025 ©2025 Association for Computational Linguistics



```
Input: Claim c, MAX_SEARCH_QUERIES, MAX_SEARCH_RESULTS_PER_QUERY
Output: Veracity Label (True or False)
```

```
/* Initialize memory bank, list of results that aren't self-contained, and the query queue */
memory_bank = []
not_self_contained_results = []
query_queue = InitialQueryGen(c)
while query_queue is not empty and number of queries made < MAX_SEARCH_QUERIES:
        search_query = POP(query_queue, 0)
        result_list = SearchEngine(search_query)
        ranked_result_list = SearchRank(search_query, result_list) [0 : MAX_SEARCH_RESULTS_PER_QUERY]

        for result in ranked_result_list:
                if SelfContainedCheck(c, memory_bank, result):
                        if DetHelpful(c, memory_bank, result):
                                Append result to memory_bank
                                if SufficientEvidence(c, memory_bank):
                                        return Classifier(c, memory_bank)
                                else:
                                        additional_queries = AdditionalQueryGen(c, memory_bank)
                                        Add additional_queries to the front of query_queue
                                        break
                        else: continue
                else: Append result to not_self_contained_results

Iterate through not_self_contained_results. If including a result in memory_bank enables classification
because there would now be sufficient evidence, return the result of classifying and terminate the
algorithm.

/* If haven't terminated at this point, do classification with the evidence collected so far */
return Classifier(c, memory_bank)
```

Figure 1: **Claim veracity classification algorithm.** The input to the algorithm is an atomic claim **c** along with two values specifying the maximum number of search queries that can be made and the maximum number of search results returned per query. The output of the algorithm is a binary label indicating the veracity of the claim. Each LLM agent is highlighted in 🟩 green .

first decomposes a text into several atomic claims and then verifies each claim individually (Hu et al., 2024; Wei et al., 2024; Song et al., 2024; Min et al., 2023). Several approaches for the latter task of verifying individual claims (which is the task that we focus on in this work), begin by retrieving evidence via web search and then feeding the evidence and the claim to a language model for final verification (Chern et al., 2023; Wang et al., 2024). Though this is sometimes effective, a shortcoming is the misalignment between this process and the process of humans when doing the task. Recent works address this through iterative evidence retrieval (Xie et al., 2025), which allows for evidence to be collected several times and only when it is considered necessary. We build on this idea with EMULATE and also incorporate the idea of iterative retrieval and verification.

3 Methodology

Emulating Human Actions. Our multi-agent framework is designed to emulate human actions. If a human were trying to verify a claim by using the Internet, they would start by making a search query that they think would be helpful, which will return many results/links. They would then select a link to click on based on the credibility of the source (which can be inferred from the URL), as well as the relevance (which can be guessed by looking at the title and snippet). After clicking on a link and reading the text, one of the following scenarios will occur:

(a) The document is self-contained and the human has sufficient information to determine the claim's veracity.

(b) The document is self-contained and the human was able to acquire knowledge that's helpful for the task, but more information is needed.

(c) The document is self-contained, but completely irrelevant.

(d) The document is not self-contained.

If scenario **(a)** occurs, the human is done with the task. If scenario **(b)** occurs, the human should retain the information acquired from the text and then think of additional search queries required for completing the task. In scenario **(c)**, the human should visit another link that was returned in the response to the initial search query. In scenario **(d)**, the human would need to formulate additional search queries to fill in the gaps. To the best of our

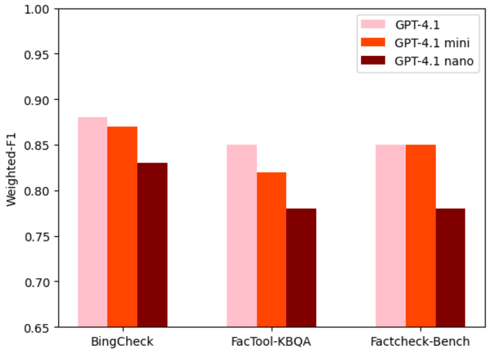


Figure 2: Results on the entire GPT-4.1 model family. The strongest/weakest model according to OpenAI is GPT-4.1/GPT-4.1-nano.

Dataset	#True	#False	Total
FacTool-KBQA	177	56	233
BingCheck	160	42	202
Factcheck-Bench	472	159	631

Table 1: Dataset statistics for FactTool-KBQA, BingCheck, and Factcheck-Bench

knowledge, our system is the first fact-checking algorithm to follow this process.

A Novel Multi-Agent Framework. Our fact-checking algorithm is shown in Figure 1, and makes use of the following LLM-powered agents: (1) **InitialQueryGen:** Generates a list of initial search queries given a claim. (2) **SearchRank:** Given a query and a list of corresponding search results (each result consists of a title, a URL, and a snippet), outputs a sorted list of the results based on relevance and credibility. (3) **SelfContained-Check:** Given a claim, the evidence set so far, and a new search result, determines if the content of the new webpage is comprehensible (i.e., if it is comprehensible, it is either self-contained or can be comprehended if you consider the information in the evidence set). (4) **DetHelpful:** Given a claim, the evidence set so far, and a new comprehensible search result, determines if the search result provides new information that isn't already mentioned in the current evidence set and if it would be helpful for veracity checking. (5) **SufficientEvidence:** Given a claim and the evidence set so far, determines if there is sufficient evidence to perform classification. (6) **Classifier:** Given a claim and the evidence set, outputs a classification label. (7) **AdditionalQueryGen:** Given a claim and the evidence set, outputs a list of search queries to enhance the existing evidence set.

Note that in Figure 1, when the algorithm encounters scenario **(d)**, it stores the result instead of making additional search queries to fill in the gaps, and walks through them at the end if it didn't terminate during the *while* loop yet, since something that was once not self-contained could become self-contained if the memory bank has changed. This design choice was made to prioritize processing self-contained evidence pieces to minimize the number of queries that need to be made.

4 Experiments

Datasets and Metrics. We evaluate EMULATE along with other systems on three datasets that each provides annotations at the level of atomic claims: FacTool-KBQA (Chern et al., 2023), BingCheck (Li et al., 2024), and Factcheck-Bench (Wang et al., 2024). FacTool-KBQA is a subset of the dataset introduced in Chern et al. (2023) for the knowledge-based QA task with 233 claims labeled as either True or False. BingCheck (Li et al., 2024) consists of atomic claims annotated with four possible labels (*supported*, *refuted*, *partially supported*, and *not supported*). We retain *supported* and *refuted* examples only and convert their labels to *True* and *False* respectively. We also only use a portion of the *supported* examples to control the class imbalance. Factcheck-Bench (Wang et al., 2024) provides 661 checkworthy claims human-annotated with either *True*, *False*, or *Unknown*. We ignore the *Unknown* examples and sample 631 claims for our experiments. See Table 1 for full dataset statistics.

To quantify performance, we report the precision, recall, and F1 scores for each class. We also provide the macro-F1 score, which aggregates the label-wise F1 scores by averaging. The weighted-F1 score is also included, which could better account for class imbalance.

Baselines. We compare our multi-agent system with four baselines: (1) FACTOOL (Chern et al., 2023), (2) FACTCHECK-GPT (Wang et al., 2024), (3) SAFE (Wei et al., 2024), and (4) FIRE (Xie et al., 2025). Note that FIRE (Xie et al., 2025) is the only baseline that was designed to take as input an atomic claim and output *True* or *False* (like EMULATE). In each of the other three baselines, checking the veracity of atomic claims is one step in the algorithm, which means that minor modifications to the corresponding open-source repositories were required to make comparisons[1].

[1] For FACTCHECK-GPT, we also modify the code to utilize *serper.dev* to obtain a maximum of 10 URLs per query.

Dataset	Method	True			False			M-F1	W-F1
		P	R	F1	P	R	F1		
BingCheck	FacTool	**0.92**	0.84	0.88	0.55	**0.71**	0.62	0.75	0.83
	FactCheck-GPT	0.91	0.8	0.85	0.48	0.69	0.56	0.71	0.79
	SAFE	0.88	0.72	0.79	0.37	0.62	0.46	0.62	0.72
	FIRE	0.91	0.87	0.89	0.58	0.69	0.63	0.76	0.84
	EMULATE	0.91	**0.96**	**0.93**	**0.79**	0.62	**0.69**	**0.81**	**0.88**
FacTool-KBQA	FacTool	**0.91**	0.84	0.87	0.59	0.73	0.65	0.76	0.82
	FactCheck-GPT	**0.91**	0.77	0.83	0.51	**0.77**	0.61	0.72	0.78
	SAFE	0.89	0.87	0.88	0.61	0.64	0.63	0.76	0.82
	FIRE	0.9	0.88	0.89	0.63	0.68	0.66	0.78	0.83
	EMULATE	0.89	**0.92**	**0.91**	**0.72**	0.64	**0.68**	**0.8**	**0.85**
Factcheck-Bench	FacTool	0.93	0.74	0.82	0.52	0.82	0.64	0.73	0.77
	FactCheck-GPT	**0.94**	0.74	0.83	0.53	**0.86**	0.65	0.74	0.78
	SAFE	0.92	0.78	0.84	0.55	0.79	0.65	0.74	0.79
	FIRE	0.93	0.81	0.87	0.59	0.81	0.68	0.78	0.82
	EMULATE	0.9	**0.89**	**0.9**	**0.7**	0.72	**0.71**	**0.8**	**0.85**

Table 2: For each claim verification system, we report the label-wise precision, recall, and F1 scores along with the Macro-F1 (**M-F1**) and Weighted-F1 (**W-F1**) scores. The best results on each dataset are shown in **bold**, while the second best results are underlined.

Dataset	Method	True F1	False F1	Weighted-F1
FacTool-KBQA	RM-SR	0.87	0.57	0.8
	RM-SCC	0.9	0.63	0.84
	EMULATE	0.91	0.68	0.85
Factcheck-Bench	RM-SR	0.88	0.68	0.83
	RM-SCC	0.88	0.66	0.82
	EMULATE	0.9	0.71	0.85

Table 3: Ablation studies on FacTool-KBQA and FactCheck-Bench. RM-SR/RM-SCC means that **SearchRank/SelfContainedCheck** were removed from EMULATE.

Implementation. For our main experiments, we employ OpenAI's GPT-4.1 model[2] with a temperature of 1 for all agents in EMULATE as well as the baseline systems. All EMULATE agents are provided with zero-shot prompts that contain instructions for the subtasks. Unless otherwise stated, MAX_SEARCH_QUERIES and MAX_SEARCH_RESULTS_PER_QUERY are set to 4 and 2 respectively. To make web searches, we invoke API calls with *serper.dev*.

5 Results

Our main results are presented in Table 2. From them, we can see that EMULATE outperforms all baselines on every dataset on 6 out of 8 metrics that we compute. Notably, **EMULATE consistently achieves the best results on both label-wise F1 scores, the macro-F1 score, and the weighted-F1 score**, which confirms the effectiveness of our novel design. We also observe that FIRE always achieves the second best results, which is likely attributed to its iterative retrieval mechanism.

To gain a better understanding of the impact that different agents have on our system, we conduct ablation studies on FacTool-KBQA and FactCheck-Bench. In particular, we quantify the effect of removing (1) **SearchRank** and (2) **SelfContainedCheck**. From Table 3, we can see that excluding **SearchRank** leads to performance degradation on both datasets (more heavily on FacTool-KBQA), which tells us that the **SearchRank** agent can effectively sort a list of search results according to the aforementioned criteria. We also find degradation on all datasets when removing the **SelfContainedCheck** agent, which reveals that the agent can effectively evaluate and filter search results.

Lastly, we run experiments on the entire GPT-4.1 model family to determine if EMULATE still works well when the underlying LLM of each agent is supplanted with a weaker model. According to Figure 2, as the underlying LLM weakens, the weighted-F1 scores decrease as well. Intuitively, weaker models are expected to be less performant on the subtasks in EMULATE, which can lead to suboptimal results; however, we can see that the performance of EMULATE when equipped with GPT-4.1-mini is sometimes close to the performance with GPT-4.1.

6 Conclusion

In this paper, we proposed a novel approach for determining the veracity of atomic claims, which is designed to emulate human actions through a multi-agent framework. Through extensive experiments, we showed that our system, EMULATE,

[2] gpt-4.1-2025-04-14

outperforms previously introduced algorithms for the task and can work well even when used with a weaker base LLM. We also reported the results from doing ablation studies, which confirmed the effectiveness of several agents.

Limitations

Evaluation of our system requires datasets that have veracity annotations at the level of atomic claims. Due to the scarcity of such datasets, we were only able to evaluate on three, and each contained less than 1,000 examples. Additionally, these datasets have a class imbalance issue (i.e., there are significantly less *False* claims than *True* claims).

Another shortcoming lies in our design choice of processing documents that aren't self-contained at the end of the algorithm. Future work should investigate other alternatives, since for some claims, it may not be possible to do claim verification without providing search results that aren't self-contained as evidence.

References

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, and 1 others. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2024. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? *arXiv preprint arXiv:2411.02400*.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *arXiv preprint arXiv:2402.07401*.

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. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 163–181.

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.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004, Toronto, Canada. Association for Computational Linguistics.

Ronit Singal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using rag and few-shot in-context learning with llms. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 91–98.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304.

Yuxia Wang, Revanth Gangi Reddy, Zain Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, and 1 others. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Zixia 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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhuohan Xie, Rui Xing, Yuxia Wang, Jiahui Geng, Hasan Iqbal, Dhruv Sahnan, Iryna Gurevych, and Preslav Nakov. 2025. FIRE: Fact-checking with iterative retrieval and verification. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2901–2914, Albuquerque, New Mexico. Association for Computational Linguistics.