Multi-LLM Agents Architecture for Claim Verification

Giuseppe Fenza[1,*,†], Domenico Furno[1,*,†], Vincenzo Loia[1,†] and Pio Pasquale Trotta[1,2,*,†]

[1]Department of Management and Innovation Systems, University of Salerno, Fisciano (SA), 84084, Italy
[2]IMT School for Advanced Studies, Lucca (LU), 55100, Italy

Abstract

The rapid spread of misinformation has made automated claim verification essential, as traditional methods struggle to keep pace. While NLP advancements and Large Language Models as GPT-4 show promise, their limitations, such as outdated knowledge, underscore the need for scalable, domain-independent solutions that integrate external resources. This work introduces a novel multi-agent architectural model designed for claim verification, achieving state-of-the-art performance on the FEVER dataset. The proposed system leverages specialized agents powered by Large Language Models (LLMs), integrated within a modular and scalable two-layered framework comprising a Reasoning Layer and a Decision Layer. Extensive experimentation demonstrates significant performance improvements, with the optimized system achieving 85.31% accuracy and 85.29% F1-Macro, outperforming traditional single-model baselines. The study also explores the impact of individual agents' contributions and highlights the effectiveness of reducing system complexity to enhance both accuracy and conclusiveness. These findings establish the potential of multi-agent systems to transform claim verification by offering robust, and flexible solutions.

Keywords

Claim Verification, Multi-Agents Systems, LLM

1. Introduction

The digital era has transformed drastically how information is produced, accessed, and shared, enabling people to stay constantly connected and updated on current events. However, this technological revolution has also brought significant challenges because, in this scenario of uncontrolled production of knowledge, it is becoming more and more difficult to distinguish reliable information from untrustworthy ones. Indeed, the proliferation of disinformation poses serious threats to public trust, decision-making, and societal stability. For those reasons and also because traditional methods are not suitable to combat the rapid spreading of disinformation across various platforms, further improvements to fact-checking and claim verification processes are crucial. Among the most promising approaches for tackling disinformation are advancements in automated fact-checking technologies, which leverage Natural Language Processing and Machine Learning techniques to analyze vast quantities of data and deliver evaluations of their veracity considering reliable sources, In [1] rely on deep learning techniques to detect fake news, combining traditional neural networks for classification tasks (e.g., Convolutional Neural Network, Multi-Layer Perceptron) and pre-trained models (e.g., BERT [2], RoBERTa [3]) to achieve better performance. For the claim verification task, Pankovska et al. [4] use strategies that refer to textual sources or knowledge bases, allowing them to make informed decisions based on evidence retrieval results. The previously mentioned solutions are typically trained on specific data, making them inadequate for use across several domains, highlighting the necessity to develop new scalable and accurate methods employable for different types of domains. Nowadays, thanks to the advancements in Natural Language Processing, Large Langauge Models (LLMs) have risen, such as GPT-4 [5] and Gemini 1.5 Pro [6], showing astounding capabilities in comprehending and generating human-like text and also in performing different tasks (e.g., translation, code programming) because of their training on

Joint National Conference on Cybersecurity (ITASEC & SERICS 2025), February 03-8, 2025, Bologna, IT
*Corresponding author.
†These authors contributed equally.
gfenza@unisa.it (G. Fenza); dfurno@unisa.it (D. Furno); loia@unisa.it (V. Loia); piopasquale.trotta@imtlucca.it (P. P. Trotta)
0000-0002-4736-0113 (G. Fenza); 0009-0007-6937-2864 (D. Furno); 0000-0003-4807-8942 (V. Loia)

massive and heterogeneous datasets. However, their knowledge might not be updated, and this could pose a significant problem for the claim verification task to obtain updated and truthful evaluations of claims. This problem can be addressed by retrieving additional external knowledge (for example, exploiting search engines or existing knowledge graphs) to integrate in a prompt to provide to an LLM or by fine-tuning the LLM to improve its performance for the claim verification task. Current research focuses mostly on employing a single large language model, which could result in a limited approach for complex tasks as the claim verification one. In this field and other complex decision-making processes, the use of multi-agent systems - based on one or more LLMs - could represent a significant advancement over the traditional reliance on a single, general-purpose large language model. These systems are designed to mimic the collaborative and specialized efforts of human teams, where each member contributes unique expertise to address a multifaceted problem effectively [7, 8]. Multi-agent systems enhance precision, scalability, and transparency by assigning specialized tasks to individual agents, enabling parallel processing and modular adaptability. This approach ensures robust, flexible, and accountable solutions, particularly in complex domains [9]. This work presents a novel architectural model for a multi-agent system for claim verification. The model consists of two layers: *Reasoning* and *Decision* layers. The *Reasoning* layer can comprise multiple agents powered by a large language model (LLM), each executing distinct tasks and using shared tools. Instead, the *Decision* layer is composed of multiple agents responsible for synthesizing other agents' responses to deliver a well-informed verdict on the claims veracity. Each agent exploits ReAct [10] and Reflexion [11] mechanisms to iteratively evaluate and adjust the execution plan by incorporating insights from previous actions and observations, addressing past errors, and enhancing the overall quality of the final outcomes. This robust architectural model for multi-agent system is designed to address the complexities of claim verification, offering scalability and flexibility by enabling the seamless addition of new agents or tasks without disrupting the overall framework. This adaptability promotes efficiency, precision, and modularity, ensuring the system remains effective in handling diverse verification challenges. Additionally, the paper details the optimization process employed to enhance the system's performance in a binary classification task using the FEVER dataset, a benchmark that provides a collection of claims paired with corresponding evidence from Wikipedia. The optimization involved analyzing various configurations, evaluating their overall performance, and examining the individual contributions of each agent to understand how they affect the final outcome. This process allowed the achievement of state-of-the-art performance on the aforementioned benchmark in a binary classification task. This study also addresses key research questions to explore the strengths and potential of the proposed approach:

- **RQ1**: How do individual agents' contributions affect the final outcome?
- **RQ2**: Does a multi-agent system outperform a single LLM and other baselines on claim verification using this architectural model?
- **RQ3**: What is the relation between the number of agents and performance?

The paper continues as follows. Section 2 presents the existing literature about the claim verification task and the use of LLM agents. The architectural model is described in Section 3. Section 4 shows the results achieved and outlines the refinement process used to enhance the accuracy of the multi-agent systems within the proposed architectural model, while also addressing the research questions. Lastly, Section 5 illustrates the limitation of our approach, while conclusions are presented in Section 6.

2. Related Works

The claim verification problem is often addressed through automated methodologies composed of three key phases, which can be tackled either individually or collectively [12]: *Claim Detection*, *Evidence retrieval*, and *Claim Validation* (or *Verification.*) Frequently, evidence retrieval and claim validation are combined into a single, integrated task. Additionally, the process of claim validation may include generating an *explanation* or *justification* for the verdict rendered on a claim.

Claim detection identifies assertions requiring fact-checking. Approaches often fine-tune models like BERT and T5 [13], integrate meta-features for classification, or verify claims against databases using ranking methods [14]. Annotated corpora, such as those for climate change [15], support this task. *Evidence retrieval* involves finding relevant information to confirm or refute claims, improving understanding and enabling comparisons with reliable sources. Techniques range from traditional information retrieval [16] to leveraging search engines [17], knowledge graphs [18, 19], or focusing on documents published before the claim was made [20]. *Claim validation* evaluates a claim's veracity using methods such as binary classification [21], referencing evidence or knowledge bases [4], or employing graph-based approaches [22]. Lately, various strategies leveraging large language models (LLMs) in claim verification have been proposed. For example, using perplexity scores from pre-trained language models on claim-evidence pairs for classification has been explored [23]. Alternatively, fine-tuning LLMs on domain-specific data can improve performance but is resource-intensive [24, 25, 19]. Also, in-context learning [26] demonstrates the effectiveness of few-shot prompting to achieve robust results without extensive training. With the advancement of LLMs, LLM agents have been utilized in diverse applications, including solving coding and math problems [8], simulating human behavior [27], and engaging in debates to address complex challenges [28], often as part of multi-agent systems. As for the claim verification and fact-checking processes, Li et al [29] propose *FactAgent*, a single fact-checking agent driven by an LLM and capable of using different external tools. Moreover, Zhao et al. [30] present a framework composed of four LLM agents, leveraging LLMs with dynamic planning for claim verification. This work introduces a scalable and flexible multi-agent architectural model for claim verification, featuring a Reasoning layer for task-specific analysis by LLM-powered agents and a Decision layer for synthesizing responses into a verdict, enhanced by iterative ReAct and Reflexion mechanisms to improve precision and adaptability.

3. Methodology

The claim verification process is a multi-step pipeline aimed at determining the veracity of a given claim by analyzing relevant evidence. This process typically involves tasks such as claim detection, evidence retrieval, and claim validation. To address the inherent complexities, we propose a multi-layered architecture (Figure 1) that employs agents powered by large language models (LLMs) to ensure scalability, adaptability, and precision. The system's agents, tools, and tasks are defined using the CrewAI framework[1]. For the underlying large language model, the open-source Gemma2-9B-IT [31] is utilized. This model, known for its high performance, is accessed and managed through the Ollama framework[2].

Our architectural model is divided into two distinct layers: Reasoning Layer and Decision Layer, each playing a specific role in processing and validating claims. The agents' workflow within the proposed architectural model and the layers are detailed below.

3.1. Architectural Design & Workflow

The *Reasoning Layer* serves as the core analytical component of the proposed architectural model. It consists of multiple specialized agents, each tasked with a specific role in the claim verification process. Each agent operates based on its predefined task, guided by a goal-oriented framework and supported by dedicated tools to perform its function effectively. The agents are designed with backstories that define their individual focus and operational logic, enabling them to approach tasks with domain-specific perspectives. Collaboration among agents is facilitated through delegation: when an agent encounters a need for additional expertise, it can delegate tasks to other agents. The responses from these delegated agents are then integrated by the delegating agent to complete its own task. This collaborative reasoning

[1] https://www.crewai.com/
[2] https://ollama.com/

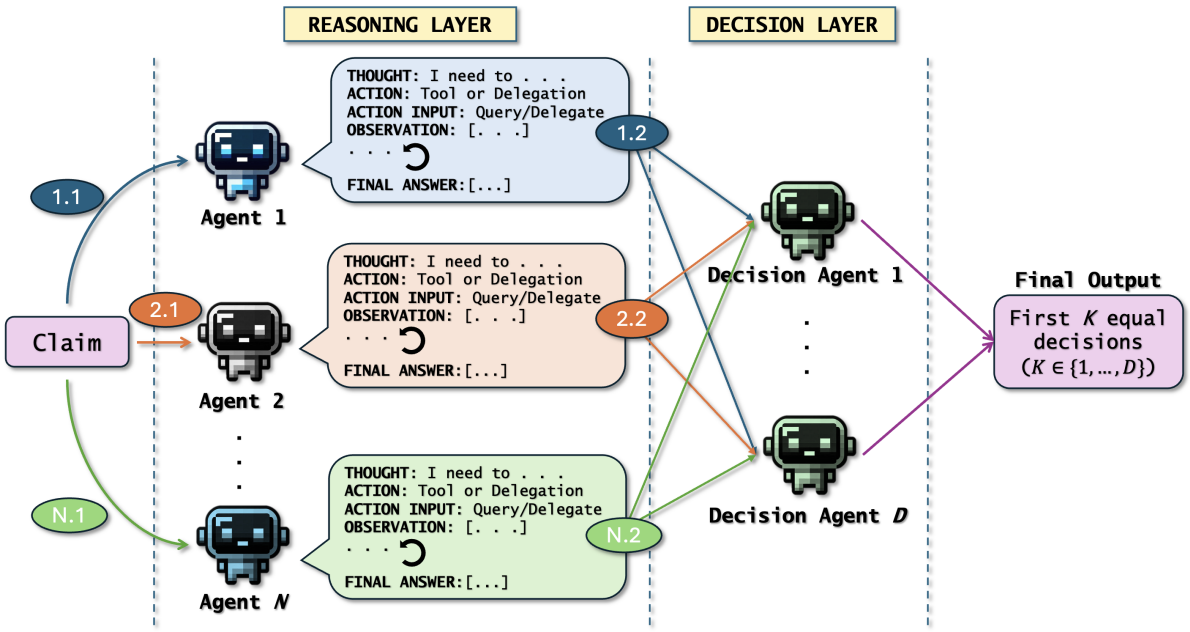

Figure 1: Proposed architectural model and workflow for an LLM-powered multi-agent system for claim verification.

is enhanced by ReAct (Reasoning and Acting) and Reflexion mechanisms, which empower the agents to iteratively refine their thoughts and actions, ensuring robust and adaptable processing of claims.

The workflow begins with the first agent in the Reasoning Layer, which receives the input claim. The agent analyzes the claim using its tools and the ReAct and Reflexion mechanisms to generate a task-specific output. Once this task is completed, the output is passed to the Decision Layer. This process continues iteratively until all agents in the Reasoning Layer have completed their tasks.

The *Decision Layer* comprises multiple Decision Agents, responsible for aggregating and synthesizing the outputs from the Reasoning Layer. These agents evaluate the coherence of the collected information and determine the final verdict on whether the claim is supported or refuted. Importantly, the Decision Agents are intentionally designed without the ability to delegate tasks or utilize tools (Figure 2). This constraint ensures that the impact of each Reasoning Layer agent on the final verdict is isolated and clear, avoiding potential biases introduced by further delegation or tool use. The final verdict is determined by considering the first K consistent outputs from the Decision Agents, where K is a tunable parameter ($K \in \{1, ..., D\}$) and D is the total number of decision agents.

Decision Agents parameters

Goal: Analyze and synthesize outputs from multiple agents to determine if the claim is supported or refuted. Label can be SUPPORTS or REFUTES.
Backstory: A critical thinker with expertise in evidence synthesis and decision-making under uncertainty. Skilled in integrating multi-agent outputs to reach a comprehensive and unbiased conclusion.
Task: Based on the outputs of *Agent 1,..., Agent N*, provide a final decision on the claim: *CLAIM*. The final decision should be based primarily on factual accuracy.
Context: Task Output Agent 1,..., Task Output Agent N
Expected Output: A final decision: SUPPORTS or REFUTES the claim.

Figure 2: Goal, backstory, task, context considered, and expected output of the Decision Agent.

3.2. Agents Configuration and Tools

The initial multi-agent system configuration within the proposed architectural model comprises four specialized agents in the Reasoning layer, each designed to address distinct aspects of the claim verification process. These agents are strategically crafted to analyze various key dimensions of claim validation (Figure 3 shows how each agent is characterized). Their specific goals are outlined below:

- **Fact-Checking Agent:** Verify the factual accuracy of the claim by identifying verifiable evidence and providing reliable sources.
- **Context Analyst Agent:** Analyze the broader context of the claim, such as its historical accuracy, its alignment with socio-political events, and its cultural significance.
- **Media-Bias Analyst Agent:** Evaluate potential biases in the claim source and identify patterns or tones indicative of manipulation or partisanship.
- **Public Sentiment Analyst Agent:** Gauge public reaction to the claim.

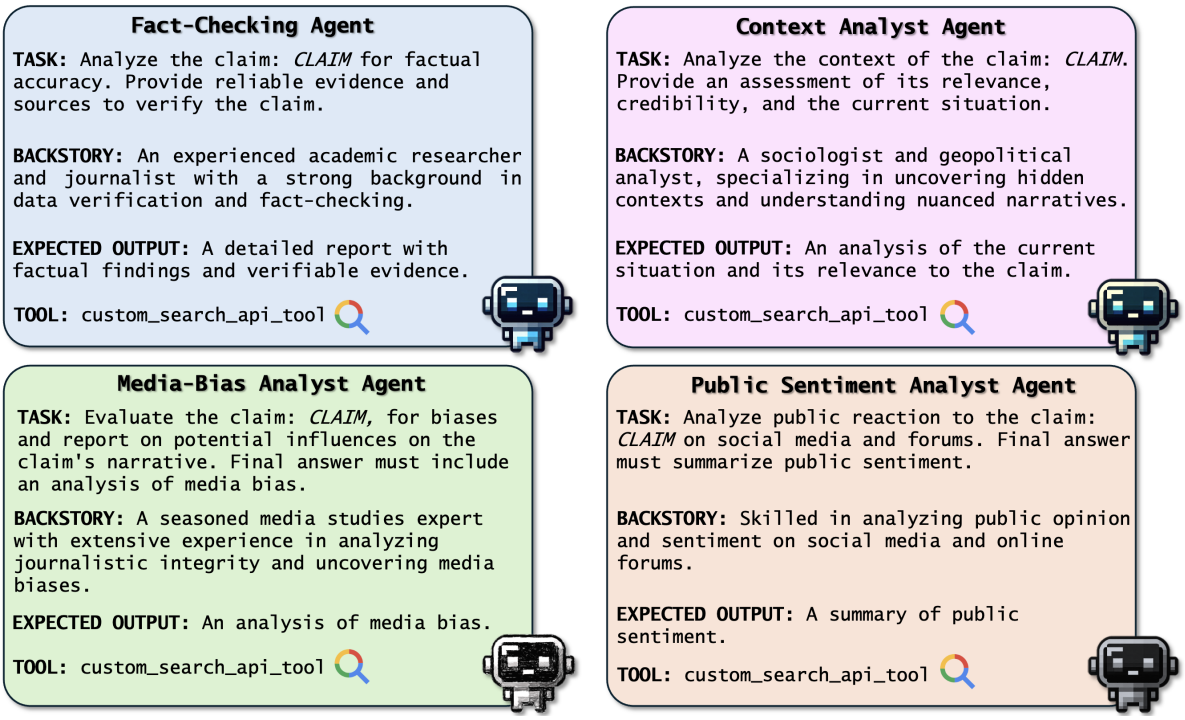

Figure 3: Detailed description of the agents considered within the system.

Each agent is based on the LLM Gemma2-9B-IT, whose temperature parameter is set to 0.1 for a more concise output, and equipped with the same tool called *custom_search_api_tool*, a research utility that leverages the Google search engine via the Google Custom Search JSON API[3]. This API enables the agents to query a Programmable Search Engine[4], facilitating efficient information retrieval for their own task. Additionally, they have the opportunity to delegate to other agents, seeking support to complete their task. Also, a maximum number of iterations for a task is set to avoid infinite loops and reduce the time needed to finish the task. However, this could also lead to uncompleted tasks, resulting in the task output *Agent stopped due to iteration limit or time limit.* This output is important as it highlights which agent fails to complete its task, providing insights into its impact on the final results. It also enables a deeper analysis of each agent's actions and contributions to the system's overall performance. Lastly, K is set to 1 in this setup, meaning the system considers the output of a single Decision Agent as the final verdict.

[3] https://developers.google.com/custom-search/v1/overview
[4] https://programmablesearchengine.google.com/controlpanel/create

The first multi-agent system setup achieves state-of-the-art performance on the FEVER dataset in a binary classification task compared to other baselines (Table 2), but further experiments have been carried out.

4. Experimentation & Results

This section outlines the various multi-agent systems' experimental setups derived from the optimization process conducted to address the previously stated research questions. Additionally, it describes the dataset used and presents the results obtained for each setup, which are then compared against different baseline models.

4.1. Dataset

To evaluate the framework, the widely-used Fact Extraction and VERification (FEVER) dataset was adopted [32]. This dataset is specifically designed for claim verification tasks, where claims are categorized into three labels: SUPPORTS, REFUTES, and NOT ENOUGH INFO, based on annotations by human experts. The claims were generated by rephrasing sentences from Wikipedia, and for each claim, the dataset includes supporting sentences extracted from relevant Wikipedia pages that annotators used during the labeling process. For this study, the FEVER Development dataset was utilized, but the evaluation focused on a two-class classification problem, considering only claims labeled as SUPPORTS or REFUTES, for a total of 13,332 claims, providing a robust foundation for assessing the system's performance.

4.2. System Optimization and Evaluation

Although the initial setup demonstrated strong performance, a deeper analysis was undertaken to explore potential improvements for the Reasoning Layer. This process aimed to answer the first research question **(RQ1)**: **How do individual agents' contributions affect the final outcome?** Table 1 presents the percentage of instances where agents failed to complete their tasks within the iteration limit, along with their contributions to correct labels and their association with inconclusive answers.

Agent	Failures (%)	Inconclusive (%)	Contributions (%)
Fact-Checking	11.82%	18.8%	63.39%
Context Analyst	8.08%	11.28%	66.07%
Media-Bias Analyst	**25.72%**	**37.59%**	**54.04%**
Public Sentiment Analyst	**42.35%**	**91.73%**	**47.09%**

Table 1
Proportion of cases where agents failed to complete their tasks, along with their contributions to correct labels and inconclusive answers.

The columns in Table 1 represent: **Failures (%)**, the percentage of tasks where the agent was unable to complete its operation within the iteration limit; **Inconclusive (%)**, the percentage of tasks where the agent was unable to complete its operation within the iteration limit and resulted in an inconclusive answer; **Contributions (%)**, the percentage of tasks completed by the agent that contributed to a *correct* label.

4.2.1. Analysis of Failures and Contributions

As shown in Table 1, the Public Sentiment Analyst and Media-Bias Analyst were the agents most frequently involved in blocked iterations, with failure rates of 42.35% and 25.72%, respectively. These agents also contributed significantly to inconclusive answers, accounting for 91.73% and 37.59% of such cases. Their relatively low contributions to correct labels (47.09% and 54.04%) further highlight

their limited effectiveness. In contrast, the Fact-Checking and Context Analyst agents demonstrated higher reliability, with failure rates of only 11.82% and 8.08%, respectively, and strong contributions to correct labels (63.39% and 66.07%). These findings suggest that the outputs of these agents had a more positive influence on the final outcomes.

4.2.2. Optimization Through System Simplification

Based on this analysis, two optimized setups were tested to improve overall performance: (1) **Multi-agent system 2**, where the most problematic agent, the Public Sentiment Analyst, was removed; (2) **Multi-agent system 3**, where both the Public Sentiment Analyst and the Media-Bias Analyst were excluded, leaving only the Fact-Checking and Context Analyst agents. Both configurations outperformed the original system, with *Multi-agent system 3* achieving the best results. This demonstrates that reducing system complexity by removing less effective agents can significantly enhance both accuracy and conclusiveness, while preserving the collaborative framework between the remaining agents.

4.3. Baselines

The multi-agent systems within the proposed architectural model for FEVER binary classification are compared with state-of-the-art approaches, particularly those based on Large Language Models. These include a perplexity-based method (PPL) [23], which uses conditional perplexity scores from pre-trained language models to classify claim-evidence pairs as either $Supports$ or $Refutes$ based on a threshold. Fine-tuned models such as $BERT - B_{ft}$ and $XLNET_{ft}$ [23] are also evaluated for binary classification. The performances of the multi-agent systems are measured using Accuracy and F1-macro metrics. Accuracy is calculated as the ratio of correctly predicted instances to the total instances, and F1-macro is the average of the F1 scores for each class, which is particularly useful for imbalanced datasets. The F1 score for a class is computed as the harmonic mean of Precision and Recall, with the F1-macro score being the average of all class F1 scores. If the Decision Agent gives an inconclusive answer (i.e., one that does not assign a single label), the instance is excluded from the final result. Once calculated, the results are presented alongside the best baseline scores.

4.4. Results

Model	Accuracy (%)	F1-Macro (%)
$BERT - B_{ft}$	52.18	38.82
$XLNET_{ft}$	49.18	48.42
$PPL_{GPT2\text{-}XL}$	73.67	71.71
Multi-agent system 1	**78.01**	**77.53**
Multi-agent system 2	**78.71**	**78.31**
Multi-agent system 3	**85.31**	**85.29**

Table 2
Performance of the diverse multi-agent systems setup within the architectural model proposed compared to other baselines.

This section answers the second research question (**RQ2**): **Does a multi-agent system outperform a single LLM and other baselines on claim verification using this architectural model?**. Indeed, Table 2 presents the performance of different models and multi-agent system setups for the claim verification task, showing the Accuracy and F1-Macro scores. The baseline models include $BERT\text{-}B_{ft}$ and $XLNET\text{-}_{ft}$, with $BERT\text{-}B_{ft}$ achieving 52.18% accuracy and 38.82% F1-Macro, while $XLNET\text{-}_{ft}$ performs slightly worse in accuracy (49.18%) but better in F1-Macro (48.42%). The $PPL\text{-}_{GPT2\text{-}XL}$ model outperforms the others with 73.67% accuracy and 71.71% F1-Macro. The multi-agent system setups show a clear improvement over these baselines, with *Multi-agent system 1* achieving 78.01% accuracy

and 77.53% F1-Macro, indicating a notable gain in performance. *Multi-agent system 2*, without an agent, achieves even better results with 78.71% accuracy and 78.31% F1-Macro. The best performing setup, *Multi-agent system 3*, with fewer problematic agents, significantly outperforms all other models with 85.31% accuracy and 85.29% F1-Macro, demonstrating the effectiveness of the multi-agent architecture in improving claim verification performance by leveraging specialized agents. These results highlight the potential of multi-agent systems, particularly in complex tasks like claim verification, where agent specialization and collaboration contribute to higher accuracy. Lastly, the third research question (**RQ3**): **What is the relation between the number of agents and performance?**, is answered considering these results and the percentage of inconclusive answers (i.e., answers different from a single label) provided by the Decision Agent in the diverse mutli-agent systems considered.

Model	Inconclusive Answers
Multi-agent system 1	10.65%
Multi-agent system 2	8.89%
Multi-agent system 3	1.8%

Table 3
Percentages of inconclusive answers for each tested setup.

As shown in Table 2, the performance of the multi-agent systems improves as the system complexity (i.e., number of agents) decreases. Specifically, the most complex setup, *Multi-agent system 1*, which involves four agents, achieves 78.01% accuracy and 77.53% F1-Macro. As we reduce the number of agents in *Multi-agent system 2*, which has three agents, performance improves slightly. The least complex system, *Multi-agent system 3*, with only two agents, achieves the highest performance with 85.31% accuracy and 85.29% F1-Macro, indicating that reducing complexity has a positive impact on performance. Moreover, when analyzing the percentage of inconclusive answers (i.e., answers different from a single label) provided by the Decision Agent, as shown in Table 3, we observe that the number of inconclusive answers decreases as the system complexity decreases. The most complex setup, provides 10.65% inconclusive answers, while *Multi-agent system 2*, with fewer agents, reduces this to 8.89%. The least complex setup, with only two agents, results in just 1.8% inconclusive answers, showing that reducing complexity leads to more conclusive results. To conclude - considering the type of agents employed, their tools and the type of interaction (i.e., the delegation) - there is an inverse relationship between system complexity in the Reasoning layer and performance, where reducing complexity from the initial setup to the third setup results in a significant performance improvement, both in terms of higher accuracy and fewer inconclusive answers. This indicates that in the proposed model, the specific reasoning tasks and contributions of individual agents led to improved performance when the number of agents was reduced. However, this outcome may not generalize to all multi-agent systems. Future research will explore more advanced reasoning models, such as debate-based approaches, to fully leverage the potential of multi-agent architectures.

5. Discussion & Limitations

The results presented in this study highlight the potential of multi-agent systems for the claim verification task, demonstrating notable improvements in accuracy and F1-macro scores compared to traditional models. However, certain limitations must be acknowledged. Firstly, the current approach was tested on a single benchmark dataset (FEVER), and the tasks assigned to some agents, such as the Public Sentiment Analyst Agent and the Media-Bias Analyst Agent, were not well-suited for this dataset. This mismatch led to reduced performance in setups that included these agents, suggesting that the system's overall effectiveness could be influenced by the nature of the dataset and the specific responsibilities assigned to agents. Moreover, the evaluation considered only a binary classification task, and extending the system to multi-class classification or other domains might reveal further challenges. The performance of the system heavily relies on the interplay between the Reasoning and

Decision layers. In the Reasoning layer, agents are tasked with specific roles, contributing specialized insights to the claim verification process. However, this specialization also introduces the risk of misalignment between the tasks of individual agents and the dataset requirements, as observed in the less effective setups. The Decision layer, while pivotal in synthesizing outputs from the Reasoning layer, faces challenges when agent outputs are ambiguous or contradictory. The percentage of inconclusive answers provided by the Decision Agent, as shown in the experiments, underscores the importance of refining the mechanisms through which this layer resolves uncertainties and ensures consistency. Also, while reducing the system's complexity improved its performance, it remains unclear whether adding more specialized agents with better cooperation could enhance outcomes. The reduced complexity in *Multi-agent system 3* highlighted the importance of efficient collaboration among agents within the Reasoning layer and the need for the Decision layer to adapt dynamically to varying levels of input complexity. For future work, several directions could be pursued. Testing a more collaborative approach between agents in the Reasoning layer, possibly through improved communication protocols or shared memory mechanisms, could lead to better coordination and enhanced results. Enhancing the Decision layer with more sophisticated aggregation and uncertainty resolution techniques could further improve the system's robustness, or increasing the parameter K of Decision Agents. Expanding the evaluation to other datasets and tasks would also provide a more comprehensive understanding of the system's versatility. Additionally, integrating advanced methods for dynamic agent task allocation could enable the system to adapt to different datasets and objectives, further improving its scalability and robustness. Despite these limitations, the proposed multi-agent system architectural model demonstrates significant promise, providing a solid foundation for future advancements in automated claim verification.

6. Conclusions

This work presents a novel multi-agent architectural model for claim verification, leveraging Large Language Models (LLMs) within a modular and adaptable framework. Specialized agents in the Reasoning layer, supported by ReAct and Reflexion mechanisms, collaborate to provide insights synthesized by a Decision layer into a final verdict. The proposed multi-agent architectural model outperformed traditional baselines, with *Multi-agent system 3* achieving 85.31% accuracy and 85.29% F1-Macro, significantly surpassing the best-performing baseline. These findings establish the potential of multi-agent systems to enhance claim verification and fact-checking processes.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Declaration of Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Samadi, M. Mousavian, S. Momtazi, Deep contextualized text representation and learning for fake news detection, Information processing & management 58 (2021) 102723.

[2] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[3] Y. Liu, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 364 (2019).

[4] E. Pankovska, K. Schulz, G. Rehm, Suspicious sentence detection and claim verification in the covid-19 domain, in: Proceedings of the Workshop Reducing Online Misinformation through Credible Information Retrieval (ROMCIR 2022), CEUR-WS, Stavanger, 2022.
[5] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).
[6] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al., Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arXiv preprint arXiv:2403.05530 (2024).
[7] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, Z. Liu, Chateval: Towards better llm-based evaluators through multi-agent debate, arXiv preprint arXiv:2308.07201 (2023).
[8] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, C. Wang, Autogen: Enabling next-gen llm applications via multi-agent conversation framework, arXiv preprint arXiv:2308.08155 (2023).
[9] C. De Maio, G. Fenza, D. Furno, T. Grauso, V. Loia, A multi-agent architecture for privacy-preserving natural language interaction with fhir-based electronic health records, in: 2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), IEEE, 2024, pp. 1–6.
[10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, arXiv preprint arXiv:2210.03629 (2022).
[11] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, S. Yao, Reflexion: Language agents with verbal reinforcement learning, Advances in Neural Information Processing Systems 36 (2024).
[12] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206.
[13] S. Du, S. D. Gollapalli, S.-K. Ng, Nus-ids at checkthat! 2022: identifying check-worthiness of tweets using checkthat5, Working Notes of CLEF (2022).
[14] S. Shaar, N. Babulkov, G. Da San Martino, P. Nakov, That is a known lie: Detecting previously fact-checked claims, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3607–3618.
[15] T. Alhindi, B. McManus, S. Muresan, What to fact-check: Guiding check-worthy information detection in news articles through argumentative discourse structure, in: Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2021, pp. 380–391.
[16] A. Soleimani, C. Monz, M. Worring, Bert for evidence retrieval and claim verification, in: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42, Springer, 2020, pp. 359–366.
[17] G. Wang, L. Chillrud, K. McKeown, Evidence based automatic fact-checking for climate change misinformation, in: International Workshop on Social Sensing on The International AAAI Conference on Web and Social Media, 2021.
[18] B. Zhu, X. Zhang, M. Gu, Y. Deng, Knowledge enhanced fact checking and verification, IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021) 3132–3143.
[19] P. P. S. Dammu, H. Naidu, M. Dewan, Y. Kim, T. Roosta, A. Chadha, C. Shah, Claimver: Explainable claim-level verification and evidence attribution of text through knowledge graphs, arXiv preprint arXiv:2403.09724 (2024).
[20] J. Chen, G. Kim, A. Sriram, G. Durrett, E. Choi, 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), 2024, pp. 3569–3587.
[21] N. Naderi, G. Hirst, Automated fact-checking of claims in argumentative parliamentary debates, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), 2018, pp. 60–65.
[22] Z. Liu, C. Xiong, M. Sun, Z. Liu, Fine-grained fact verification with kernel graph attention network, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7342–7351.

[23] N. Lee, Y. Bang, A. Madotto, P. Fung, Towards few-shot fact-checking via perplexity, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 1971–1981.
[24] S. Vaghefi, V. Muccione, C. Huggel, H. Khashehchi, M. Leippold, Deep climate change: A dataset and adaptive domain pre-trained language models for climate change related tasks, in: NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning, 2022.
[25] T.-H. Cheung, K.-M. Lam, Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking, in: 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2023, pp. 846–853.
[26] X. Zhang, W. Gao, Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method, in: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 996–1011.
[27] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, M. S. Bernstein, Generative agents: Interactive simulacra of human behavior, in: Proceedings of the 36th annual acm symposium on user interface software and technology, 2023, pp. 1–22.
[28] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, S. Shi, Z. Tu, Encouraging divergent thinking in large language models through multi-agent debate, arXiv preprint arXiv:2305.19118 (2023).
[29] X. Li, Y. Zhang, E. C. Malthouse, Large language model agent for fake news detection, arXiv preprint arXiv:2405.01593 (2024).
[30] X. Zhao, L. Wang, Z. Wang, H. Cheng, R. Zhang, K.-F. Wong, Pacar: Automated fact-checking with planning and customized action reasoning using large language models, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, pp. 12564–12573.
[31] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al., Gemma 2: Improving open language models at a practical size, arXiv preprint arXiv:2408.00118 (2024).
[32] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: NAACL-HLT, 2018.