Agent-based Automated Claim Matching with Instruction-following LLMs

Dina Pisarevskaya and Arkaitz Zubiaga
Queen Mary University of London, UK
{d.pisarevskaya, a.zubiaga}@qmul.ac.uk

Abstract

We present a novel agent-based approach for the automated claim matching task with instruction-following LLMs. We propose a two-step pipeline that first generates prompts with LLMs, to then perform claim matching as a binary classification task with LLMs. We demonstrate that LLM-generated prompts can outperform SOTA with human-generated prompts, and that smaller LLMs can do as well as larger ones in the generation process, allowing to save computational resources. We also demonstrate the effectiveness of using different LLMs for each step of the pipeline, i.e. using an LLM for prompt generation, and another for claim matching. Our investigation into the prompt generation process in turn reveals insights into the LLMs' understanding and handling of the claim matching task.

1 Introduction

As part of an automated fact-checking pipeline, the claim matching (CM) component determines if two claims (factual statements) can be verified using the same piece of evidence or fact-check (Barrón-Cedeño et al., 2020; Shaar et al., 2020; Zeng et al., 2021; Guo et al., 2022; Pikuliak et al., 2023; Panchendrarajan and Zubiaga, 2024). The task has attracted increasing interest in recent years, as identifying matching claims can facilitate the subsequent efforts in claim verification by reducing the number of distinct claims to deal with.

In Pisarevskaya and Zubiaga (2025), we introduced CM as a binary classification task, proposing state-of-the-art (SOTA) few-shot learning results with four large language models (LLMs), based on hand-crafted prompt templates and few-shot examples, without automated prompt engineering. We aim to further investigate and develop the proposed instruction-following LLMs application for CM, overcoming this limitation. We propose, for the first time, a novel agent-based approach specifically for CM, which outperforms these SOTA results for CM as well as results after SOTA prompt tuning.

The main contributions of our work are:

- We investigate automated prompt engineering methods for CM, searching for best few-shot examples and prompt templates;
- To automate the CM task, we propose an LLM agent-based pipeline to generate specific CM prompts and then use them for the task, comparing performance against SOTA CM approach and the SOTA prompt tuning method;
- We study the LLMs' understanding of CM, revealing which prompts work better and why.

Our experiments reveal that LLMs are effective in identifying matching claims when the prompt provided focuses on the identification of claims describing the same event, topic or idea. We also find that LLMs of smaller size can be as effective as larger ones in creating prompts that perform well, suggesting a way to save computational resources in the prompt generation process by leveraging smaller LLMs.

2 Related Work and Motivation

Claim matching is useful for identifying claims that can be fact-checked together and hence for avoiding duplication of work during fact-checking. It is addressed as a ranking task (Shaar et al., 2020; Kazemi et al., 2021, 2022; Shaar et al., 2022) or, more recently, as a classification task (3 classes recognising textual entailment) (Choi and Ferrara, 2024a,b). Pisarevskaya and Zubiaga (2025) investigated manually generated prompt templates with a single user instruction by framing the task as a paraphrase detection, claim matching or natural language inference task, to assess the applicability of closely related tasks to CM.

LLM agents are able to interact, complete tasks, draw conclusions (Zhao et al., 2024; Wang et al.,

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2405–2414
December 20-24, 2025 ©2025 Association for Computational Linguistics

2024; Li, 2025). LLM multi-agent systems use capabilities of single agents, enhancing their collaboration on complex tasks (Guo et al., 2024; Han et al., 2025; Liang et al., 2024). Inspired by Chan et al. (2023) and Fang et al. (2025), we apply a pipeline interaction between two agents: the output of the first agent becomes the next agent's input. But, for CM, we suggest that the final label depends only on the second agent.

Automated prompt engineering allows optimising prompts by reducing human effort (Liu et al., 2021; Schulhoff et al., 2025). We investigate how an automated choice of few-shot examples for a CM prompt can lead to improved performance.

Generating prompts with LLMs is an emergent approach.Reynolds and McDonell (2021) describe meta-prompting that seeds a LLM to generate prompts. In (Zhou et al., 2023), one LLM generates instruction candidates for various tasks, evaluated with another LLM. Ye et al. (2024) suggest to initialise the prompt optimisation with LLMs either with pre-existing human-written prompts or with a batch of examples, asking to create an instruction for them. Then, an LLM can continuously propose new and potentially better prompts. Unlike this, our agentic system exists in the offline mode, to better understand the LLMs understanding of CM: the first agent generates prompts, the second agent uses them, then we evaluate the results. We are the first to explore automated prompt generation and engineering for claim matching.

Prompt-tuning is an automated method to fine-tune only a small number of model parameters for a task (Lester et al., 2021; Liu et al., 2022; Xiao et al., 2023; Yang et al., 2025). This SOTA is compared to our approach that includes automated prompt generation and few-shot learning.

3 Dataset and Experimental Setup

3.1 Dataset

We use the ClaimMatch dataset (Pisarevskaya and Zubiaga, 2025), which is based on Nakov et al. (2022). It is the only benchmark dataset for CM as a binary classification task. For consistency and comparability of our results with the previous SOTA few-shot results from Pisarevskaya and Zubiaga (2025), we used the same train/test data splitting. We took the same explicit test set of this dataset that comprises 500 matching and 500 not matching claim pairs. We started our experiments using the few-shot train examples listed in Pisarevskaya and Zubiaga (2025). For prompt tuning, the remaining 1,682 claim pairs were taken (with train & validation split 0.8 & 0.2).

3.2 Experimental Setup

We chose two open-access instruction-following LLMs for our experiments: Mistral and Llama. To compare the proposed approach with the previous SOTA in the same settings and to ensure reproducibility, we implemented the same model versions: Mistral-7B-Instruct-v0.3 (Mistral) and Llama-3-8B-Instruct (Llama) (max. new tokens 400). The same models were used for prompt tuning (5 epochs, AdamW optimiser, lr 3e-2) with PEFT (Mangrulkar et al., 2022). For prompt generation, we use two bigger models: Mistral-Small-24B-Instruct-2501 (Mistral-b) and Llama-3.3-70B-Instruct (Llama-b), both downloaded with 4-bit quantization to fit the GPU memory 40GB.

4 Methodology

First, we describe experiments on automated choice of few-shot examples for a prompt, to define which train examples to use. After that, we propose the pipeline of LLM agent-based few-shot CM that consists of two parts: automated prompt generation, and claim pairs classification if they match.

Choosing few-shot examples for a prompt. The three best hand-crafted prompt templates from Pisarevskaya and Zubiaga (2025) were taken as a basis: CM-t, PD-t, and NLI-t - the best template for claim matching, as well as the best templates for paraphrase detection and natural language inference suitable for claim matching task, respectively (see Table 3 in Appendix A). We kept the focus on structured prompts with 10 balanced few-shot examples. We examined three options of choosing few-shot instances for a prompt and compared them to the previous SOTA results with their manual choice: (1) random choice: 10 random claim pairs (5 positive and 5 negative class examples); (2) choice based on highest/lowest semantic similarity scores: developing the negative examples selection method for the train set in Kazemi et al. (2022), 5 claim pairs with highest semantic similarity scores were selected as positives, and 5 claim pairs with lowest semantic similarity scores were chosen as negatives; (3) choice with borderline semantic similarity scores added: for both positives and negatives, we took 2 examples with the highest and with the lowest scores for their class, and 1 example with the average semantic similarity score for their class, finally obtaining 5 positives and 5 negatives. We used the All-MiniLM-L6-v2 model for semantic similarity.

Generating prompts with LLMs. As an initial step, a user prompt that includes few-shot examples is given to an LLM. It explicitly mentions that the examples contain statement pairs that match or not, without any further details as to how the claim matching task should be tackled. The specific system prompt was added too: "You are a large language model. Create the best prompt for you for the described task." Provided with the same 10 few-shot examples used in other experiments, the models were aimed to generate a new prompt for a large language model for the task described in the examples. We aimed to find out what the LLM chooses as parts of the prompt for CM task.

Firstly, we conducted extensive series of experiments with each of our four models used for prompt generation: the wide initial range of prompts was generated and evaluated. Then we dropped, for each model, similar prompts with similar results. Prompts created with Mistral, if almost similar in their wording, were merged into one prompt (taking only one of them for further experiments). Prompts generated with Llama were also not always diverse: the model suggests a definition for the CM task — whether two statements describe the same event / topic / situation / idea / issue / concept. Hence, we also took only one of such prompts, if they showed similar results. For example, we do not report results for the prompt "Do the two statements describe the same event or topic?" generated with Llama, as it is similar to the prompt "Do the two statements describe the same event, topic, situation, or issue?" by the same model, included. In Mistral classification results, with both prompts, the F1 score was the same. We also generated prompts with bigger models Llama-b and Mistral-b, to make use of a more capable LLM to generate a prompt that will then be used by a smaller LLM, to save resources. We filtered them in the same way.

After a careful investigation, we finally selected the 20 most representative and diverse prompts (5 prompts generated with each model), that illustrate trends for each model. This allows us to analyse the diversity of prompts and its correlation with performance, and helps us delve into understanding the CM task and how LLMs process it. The resulting prompts are shown in Table 4 and Table 5 in Appendix A.

Claim matching with LLMs. To assess if the proposed prompts are suitable for CM and can be generalised to other models, we ran few-shot experiments with the prompts created with the models (with 10 few-shot examples added), handling CM as a binary classification task and using four setups: (1) Mistral with Mistral prompts; (2) Mistral with Llama prompts; (3) Llama with Llama prompts; (4) Llama with Mistral prompts. We also investigated if bigger LLMs are better agents to generate a prompt for a smaller LLM from the same model family: if prompts generated with a bigger Mistral-b model are suitable for a smaller Mistral model, and if prompts generated with a bigger Llama-b model are suitable for a smaller Llama model. Results are compared to the results with the overall best CM-t, PD-t and NLI-t prompt templates (see Appendix A) from Pisarevskaya and Zubiaga (2025). To compare the proposed agent-based method for automated prompts generation with the SOTA prompt tuning method, we implemented it for Mistral and Llama models, based on CM-t, PD-t, and NLI-t prompt templates.

5 Experiment Results

Few-shot examples selection. Results in Table 1 demonstrate that, for CM, there is no common pattern in prompt engineering techniques for different models, demonstrating how train examples for the few shot should be selected. For Mistral, original hand-crafted examples in the few shot still get better results than random, sorted and borderline examples. Sorted examples yield the highest scores with CM-t and PD-t templates for Mistral, but NLI-t template works better with random examples (almost outperforming SOTA for CM). Option with borderline examples does not help this model understand the task. However, for Llama all three approaches outperform the SOTA results, with random examples getting the highest scores for CM-t and PD-t templates (but not for NLI-t template as with Mistral). The choice of few-shot examples is model-specific. Hence, for the next step we continue using hand-crafted few-shot examples.

Claim matching with LLM-generated prompts. However, Table 2 shows promising results for agent-based prompt generation. Although Mistral continues to demonstrate better scores than Llama, as in the SOTA few-shot experiments, we can highlight: for both models, Mistral prompts

	Mistral		Llama	
Method	F1, %	Acc., %	F1, %	Acc., %
CM-t template				
rand	85.5	85.5	83.7	84.0
sort	86.0	86.0	79.3	80.0
bord	84.7	84.7	78.3	79.0
SOTA	90.6	90.6	77.6	78.3
PD-t template				
rand	93.1	93.1	84.0	84.0
sort	94.1	94.1	79.4	79.8
bord	93.1	93.1	83.3	83.3
SOTA	95.0	95.0	60.0	64.5
NLI-t template				
rand	88.2	88.2	67.8	70.3
sort	86.2	86.2	86.4	86.4
bord	85.8	85.8	81.6	81.9
SOTA	88.3	88.3	52.8	59.1

Table 1: Selecting few-shot examples for a prompt: random, sorted, borderline and SOTA approaches.

do not improve over SOTA. But, similarly, both models yield better results using Llama prompts, outperforming their SOTA (especially for Llama). Mistral gets the best score with L4: f1 & accuracy 96.9 (compared to f1 & accuracy 95.0 for its SOTA result with PD-t). Llama gets the best score with L2: f1 & accuracy 94.3, greatly improving its SOTA few-shot results with various prompt templates. For Mistral, L2 is the second-best Llama prompt, but it still outperforms SOTA with f1 & accuracy 95.3. We can conclude that this prompt - “Identify whether the two statements are describing the same event, topic, or idea, to determine if the statements match or not match.” – reveals in the most detailed way the essence of the CM task, which corresponds to the human understanding in the gold labels of the dataset: the same event, topic, or idea in two claims would help detect if they can be verified with the same piece of evidence, and find out if they match. Although L1 and L5 prompts are rather similar, results for both Mistral and Llama are better with the L1 prompt. A possible explanation is that the same event, topic, situation, or issue (L1) seems to be more important for CM than the same event, idea, or concept (L5). This explains why L2 demonstrates good scores for both models: it incorporates the same event and topic, as well as the request to match claims. It is interesting that longer prompts with more detailed explanations do not get better results, see e.g. L3 and M5 results. Finally, we can conclude that prompts proposed with different LLMs contain similar core, essential statements reflecting the LLMs understanding of CM: matching claims refer to the same event. It is close to PD-t template, explaining its high performance.

Bigger models do not understand the task better than the smaller models: results with their generated prompts are not better than with prompts from a smaller model (Llama). As for Mistral, only two prompt templates (Mb1 and Mb4) yield better results than SOTA for CM. Prompts generated with a bigger Llama-b model do not essentially differ much in their template pattern and core requirements from the prompts from a smaller Llama model. But all prompts from a bigger Mistral-b model have a more specific structured template pattern compared to Mistral, with clear guidelines about more details, placed after the core requirements about the same or similar information in two statements. It improved the performance specifically for Mistral model (from the same models family), but still did not outperform Mistral results with prompts from the smaller Llama. A smaller model can be used as an agent to generate a claim matching prompt, saving computational resources. Prompts generated with one LLM can work well for another LLM: Llama is better in creating prompts, then they should be passed to Mistral which is better in CM classification. Our approach also outperforms the SOTA prompt tuning approach with the best hand-crafted prompt templates (Mistral with L4 performs better than with PD-t template after prompt tuning). Using one LLM to generate CM prompts for another LLM outperforms SOTA results for the task and saves time and resources.

Claim matching task is specific because it is simultaneously similar but not the same to multiple other tasks, i.e. paraphrase detection and natural language inference, and may incorporate some of their approaches and hand-crafted prompt templates (Pisarevskaya and Zubiaga, 2025), but is more complicated and diverse. We further investigate this unique CM specifics, revealing that LLM-generated prompts for CM also overlap with prompts from close tasks.

Error Analysis. *The “same event" requirement in LLMs prompts is not fully reliable*. Both Mistral and Llama yield worse results with Mistral prompts. As class labels are mostly followed with explanations, we can conclude that there are two main types of errors: causing false positives (trying to

Prompts generated with Llama models								**SOTA approaches**		
Llama-generated					**Llama-b-generated**			**Hand-crafted prompts**		
	Llama		**Mistral**			**Llama**		Setup	F1, %	Acc., %
Prompt	F1, %	Acc., %	F1, %	Acc., %	Prompt	F1, %	Acc., %	L CM-t	77.6	78.3
L1	89.9	90.0	95.0	95.0	Lb1	59.7	64.7	L PD-t	60.0	64.5
L2	**94.3**	**94.3**	95.3	95.3	Lb2	62.6	64.9	L NLI-t	52.8	59.1
L3	79.3	79.4	94.9	94.9	Lb3	79.2	79.5	M CM-t	90.6	90.6
L4	76.6	77.0	**96.9**	**96.9**	Lb4	**88.6**	**88.7**	M PD-t	**95.0**	**95.0**
L5	88.4	88.4	94.6	94.6	Lb5	88.4	88.4	M NLI-t	88.3	88.3
Prompts generated with Mistral models										
Mistral-generated					**Mistral-b-generated**			**Prompt tuning**		
	Llama		**Mistral**			**Mistral**		Setup	F1, %	Acc., %
Prompt	F1, %	Acc., %	F1, %	Acc., %	Prompt	F1, %	Acc., %	L CM-t	57.4	59.0
M1	76.8	77.5	82.9	83.3	Mb1	95.1	95.1	L PD-t	77.3	77.4
M2	54.3	61.0	**92.5**	**92.5**	Mb2	92.6	92.6	L NLI-t	59.0	62.5
M3	68.7	69.1	87.9	88.0	Mb3	91.0	91.0	M CM-t	89.0	89.1
M4	**77.6**	**77.6**	89.9	89.9	Mb4	**96.2**	**96.2**	M PD-t	**96.4**	**96.4**
M5	61.7	63.4	85.8	85.8	Mb5	93.0	93.0	M NLI-t	82.2	82.7

Table 2: Few-shot performance with generated prompts (left and center). SOTA performance (right).

find a logical connection between unrelated claims in the negative class examples) and false negatives (if two statements in a claim pair vary in some non-substantial, or significant details, a model creates logical connections, leading to wrong conclusions). Prompt templates M2 and M3 can make a model yield a negative class label, if two claims vary in some non-substantial details. Hence, "the same event" is not a fully reliable and sufficient marker to detect if two claims in a pair match. Llama prompts, generally, contain more additional requirements than Mistral prompts: it is not only the same event, but can also be the same topic, situation, issue, idea, or concept. There are examples where such requirements are too strict: "a similar, but not the exact same event". But such broad requirements can still help reduce the number of false negatives: "the same but from different perspectives".

LLMs prompts assume that contradictory claims cannot match. In the human understanding of CM, two claims can match if their evidence is the same, even if they contain some contradictory information. Hence, consistency between two claims, proposed as a marker in M3 and, especially, M4 and L3 prompts, does not fully correspond to CM. Mistral (with M3) and Llama (with M4) explain their false negative class labels for the same claim pair: "No, the event or situation described in 1 ([...] granting a wish that 'Friends' would stay on Netflix in 2019) is not consistent with the event or situation described in 2 (Netflix announcing that 'Friends' would no longer be available [...] after the end of 2018)" and "The statements are contradictory. 1 states that 'Friends' will still be available on Netflix [...] throughout 2019, while 2 states [...] the show would no longer be available after the end of 2018". Simple positive class examples with some variations in details are usually classified correctly, but major variations can lead to misclassification.

6 Conclusion

We propose a pipeline for agent-based few-shot CM that incorporates a novel approach to automated prompt generation with LLMs. LLM-generated prompts demonstrate that they understand the specifics of the CM task, and with these prompts two LLMs outperform SOTA, based on hand-crafted prompts. We find that a prompt that considers matching claims as those describing the same event, topic or idea performs best, and we show that there is still room for improvement by incorporating additional markers.

Limitations

It should be highlighted that we do not aim to compare the results of bigger and smaller models in these experiments. Two bigger models used for prompt generation are not only of larger size, they are also from more recent generations than the smaller models we use, which could explain their better quality. On the other hand, they are applied with quantization, which could reduce their quality. Our aim is to check if prompts, generated with bigger models of the same type, would be useful for smaller models.

We intentionally provided in the initial prompt, provided to LLMs, that the examples contain statement pairs that match or do not match, without giving any additional details, or including the definition that matching claims can be verified with the same evidence, or fact-check. Firstly, our aim was to enhance LLMs generated prompts, based on their understanding of the claim matching task (instead of providing human-crafted prompts, that would define claim matching, to them), and evaluate LLMs performance with them. Secondly, claim matching task is closely connected to other tasks in the fact-checking pipeline, but it is a specific task that does not require any fact checking of the claims provided. In the series of experiments, we found out that, if manually created prompts are not carefully curated, LLMs can wrongly address claim matching as the fact-checking task and predict the veracity of two claims instead of the output label if they match / not match. Providing the human definition of claim matching, that incorporates fact-checking purposes, could encourage such model's behaviour. However, as LLMs understanding of the task has its own limitations, further experiments are needed to define better prompts - for example, using step-by-step-reasoning to let a LLM improve prompts in multiple iterations. In all Mistral experiments with Llama prompts, recall for the negative class, and precision for the positive class are significantly higher than for another class. An explanation could be that too "strict" "the same event" requirement in a prompt lead to a significant number of false negatives, where matching claims are marked as not matching, as they vary in some details. On the other hand, a model does not find any non-existent logical connection between two definitely not matching claims (reducing the number of false positives). Step-by-step-reasoning could further improve this issue.

While we kept, for the consistency, the same number of few-shot examples for a prompt, as in the SOTA experiments, further options to automate a choice of few-shot examples for a prompt should be studied, such as their ordering (Lu et al., 2022; Zhao et al., 2021), rephrasing (Yang et al., 2024), prompt and hyperparameter selection such as the number of labeled examples (Perez et al., 2021), or choice of different examples for different prompts.

Ye et al. (2024) suggest that a LLM can continuously propose new potentially better prompts. Unlike this, our agentic system exists in the offline mode, to better understand the LLMs understanding of CM as this task can be referred to a few related tasks: the first agent, initialised with a basic prompt and a batch of examples, generates instructions for them, the second agent uses them, then we evaluate the results. Continuos prompt improvement methods, as well as adding classification evaluation with LLMs as the third component of the agent-based CM pipeline, should be investigated in further research.

We also acknowledge that more research should be done to improve the generated prompts with various prompt engineering techniques (such as chain of thought), as well as test this approach on various datasets and in the multilingual setups. Such LLM-proposed markers of matching claims, as same event and consistency of two claims, can limit the task understanding, and should be further investigated.

References

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of checkthat!2020: Automatic identification and verification of claims in social media. *Preprint*, arXiv:2007.07997v1.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shan Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *ArXiv*, abs/2308.07201.

Eun Cheol Choi and Emilio Ferrara. 2024a. Automated claim matching with large language models: Empowering fact-checkers in the fight against misinformation. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1441–1449, New York, NY, USA. Association for Computing Machinery.

Eun Cheol Choi and Emilio Ferrara. 2024b. Fact-gpt: Fact-checking augmentation via claim matching with llms. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 883–886, New York, NY, USA. Association for Computing Machinery.

Dengzhao Fang, Jipeng Qiang, Xiaoye Ouyang, Yi Zhu, Yunhao Yuan, and Yun Li. 2025. Collaborative document simplification using multi-agent systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 897–912, Abu Dhabi, UAE. Association for Computational Linguistics.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8048–8057. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

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

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. 2025. Llm multi-agent systems: Challenges and open problems. *Preprint*, arXiv:2402.03578.

Ashkan Kazemi, Kiran Garimella, Devin Gaffney, and Scott A. Hale. 2021. Claim matching beyond English to scale global fact-checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4504–4517, Online. Association for Computational Linguistics.

Ashkan Kazemi, Zehua Li, Veronica Perez-Rosas, Scott A. Hale, and Rada Mihalcea. 2022. Matching tweets with applicable fact-checks across languages. *Preprint*, arXiv:2202.07094.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinzhe Li. 2025. A review of prominent paradigms for LLM-based agents: Tool use, planning (including RAG), and feedback learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9760–9779, Abu Dhabi, UAE. Association for Computational Linguistics.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55:1 – 35.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Preslav Nakov, Giovanni Da San Martino, Firoj Alam, Shaden Shaar, Hamdy Mubarak, and Nikolay Babulkov. 2022. Overview of the CLEF-2022 CheckThat! lab task 2 on detecting previously fact-checked claims. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum*.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *NeurIPS*.

Matúš Pikuliak, Ivan Srba, Robert Moro, Timo Hromadka, Timotej Smoleň, Martin Melišek, Ivan Vykopal, Jakub Simko, Juraj Podroužek, and Maria Bielikova. 2023. Multilingual previously fact-checked claim retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16477–16500, Singapore. Association for Computational Linguistics.

Dina Pisarevskaya and Arkaitz Zubiaga. 2025. Zero-shot and few-shot learning with instruction-following LLMs for claim matching in automated fact-checking. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9721–9736, Abu Dhabi, UAE. Association for Computational Linguistics.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, and 12 others. 2025. The prompt report: A systematic survey of prompt engineering techniques. *Preprint*, arXiv:2406.06608.

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2022. The role of context in detecting previously fact-checked claims. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1619–1631, Seattle, United States. Association for Computational Linguistics.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18.

Yao Xiao, Lu Xu, Jiaxi Li, Wei Lu, and Xiaoli Li. 2023. Decomposed prompt tuning via low-rank reparameterization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13335–13347, Singapore. Association for Computational Linguistics.

Adam Yang, Chen Chen, and Konstantinos Pitas. 2024. Just rephrase it! uncertainty estimation in closed-source language models via multiple rephrased queries. *Preprint*, arXiv:2405.13907.

Xianjun Yang, Wei Cheng, Xujiang Zhao, Wenchao Yu, Linda Ruth Petzold, and Haifeng Chen. 2025. Position really matters: Towards a holistic approach for prompt tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8501–8523, Albuquerque, New Mexico. Association for Computational Linguistics.

Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. 2024. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 355–385, Bangkok, Thailand. Association for Computational Linguistics.

Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Appendix

CM-t template
A Matches to B. Correct? Answer: [yes/no]
PD-t template
A. B. Question: Do A and B both refer to the same event? Yes or no? Answer: [yes/no]
NLI-t template
Suppose A. Can we infer that B? Yes or no? Answer: [yes/no]

Table 3: Hand-crafted prompt templates.

The best performing hand-crafted claim matching (CM-t) prompt template, as well as the best performing hand-crafted paraphrase detection (PD-t) and natural language inference (NLI-t) prompt templates, suitable for the claim matching task, from Pisarevskaya and Zubiaga (2025), used in the experiments, are provided in Table 3.

The full prompts generated with the models are presented in Table 4 and Table 5. The prompts were generated with smaller models (Mistral, Llama) and bigger models (Mistral-b, Llama-b). To use these prompts for claim matching classification with LLMs, they were processed for each LLM to adhere to its specific requirements to a prompt template. 10 few-shot examples were added to each prompt.

Prompts generated with Mistral
1. In the next example, do statements 1 and 2 both discuss or mention the same event or phenomenon?
2. Determine if Statement 1 and Statement 2 describe the same event or not. If they describe the same event, the answer should be "yes," otherwise it should be "no."
3. Is the event or situation described in Statement 1 consistent with the event or situation described in Statement 2? If yes, then the correct answer should be "yes." If not, then the correct answer should be "no."
4. Given two statements, determine if the information provided in both statements is consistent or contradictory. State whether the statements Match or Do Not Match.
5. Given two statements, determine if they are logically consistent or contradictory. The answer should be in the format of "match" or "not match". For example, if two statements are about the same event, person, or object, and they agree, the answer would be "match". If they disagree, the answer would be "not match". The statements should be factual and verifiable, and no assumptions should be made about the intention or interpretation of the statements.
Prompts generated with Mistral-b
1. You will be given two statements and your task is to determine if they match or not. To do this, consider the following guidelines: 1. **Content**: Check if the main subjects, events, or facts described in both statements are the same. 2. **Context**: Consider the context and timeframe of the statements. They should be related to the same event or topic. 3. **Source**: Ignore the source or author of the statements, as they might be paraphrased or reported differently. 4. **Negation**: Be mindful of negations in the statements. A statement and its negation cannot match. 5. **Irrelevant Details**: Ignore minor details or differences in wording that do not affect the core meaning of the statements. Based on these guidelines, determine if the two statements match or not. Answer with a simple "yes" or "no".
2. You are given two statements and your task is to determine if they match or not. The statements may come from various sources and cover a wide range of topics. Your goal is to assess whether the information conveyed in both statements is consistent and related, or if they are unrelated or contradictory. #### Instructions: 1. Read both statements carefully. 2. Analyze the content and context of each statement. 3. Determine if the statements match in terms of the information they present. 4. Output "yes" if the statements match, and "no" if they do not match.
3. You are given two statements and your task is to determine if they match or not. The statements may come from various sources and cover a wide range of topics. Your goal is to assess whether the information conveyed in both statements is essentially the same or if they are different. #### Instructions: 1. Read both statements carefully. 2. Compare the key information, events, and details mentioned in each statement. 3. Determine if the statements convey the same information or if there are significant differences. 4. Output "yes" if the statements match and "no" if they do not.
4. You are a large language model trained to determine whether two statements match or not match. You will be given two statements and your task is to assess their relevance to each other. The statements may come from various sources and discuss different topics. Your role is to identify if the statements are related, as indicated by whether they refer to the same event, person, or topic, and provide a "yes" if they match and a "no" if they do not. ### Guidelines: 1. **Focus on Content**: Pay attention to the main subjects, events, and key details mentioned in each statement. 2. **Contextual Relevance**: Determine if the statements are discussing the same context or incident. 3. **Specific Details**: Match specific details such as names, dates, locations, and actions described in the statements. 4. **Logical Consistency**: Ensure that the statements logically align if they are meant to describe the same event or topic.
5. You will be given two statements, labeled Statement 1 and Statement 2. Your task is to determine if the two statements match or not match. To do this, consider the following guidelines: 1. **Matching Statements**: Two statements match if they convey the same or very similar information, even if the phrasing is different. They should agree on the key facts and events described. 2. **Non-Matching Statements**: Two statements do not match if they describe different events, people, or outcomes, or if they present contradictory information.’, ’Please provide your answer as either "yes" (for matching statements) or "no" (for non-matching statements).

Table 4: Mistral-generated prompts for claim matching.

Prompts generated with Llama
1. Do the two statements describe the same event, topic, situation, or issue?
2. Identify whether the two statements are describing the same event, topic, or idea, to determine if the statements match or not match.
3. Given two statements, determine whether they are describing the same event, idea, or concept, or if they are unrelated. Please provide a binary answer: ’yes’ if the statements match, or ’no’ if they do not match. The statements may be from different sources, and the model should rely on its understanding of the content to make a decision. Examples of matching statements include identical events or ideas, while non-matching statements may be unrelated topics, contradictory information, or different perspectives on the same issue.
4. Given two statements, determine if they are describing the same event, topic, or idea. Please indicate whether the statements ’match’ or ’do not match’ based on the information provided. Use the examples below as a reference for understanding the format and tone of the statements. Please respond with a simple ’yes’ or ’no’ to indicate whether the statements match or not match.
5. Are the two statements describing the same event, idea, or concept?
Prompts generated with Llama-b
1. Compare the information in Statement 1 and Statement 2 to determine if they convey the same information or describe the same event. Consider the context, facts, and details presented in each statement. Output ’yes’ if the statements match and ’no’ if they do not match.
2. Compare the information presented in Statement 1 and Statement 2. Determine if the main claims, events, or facts described in both statements are consistent with each other. If the statements convey the same overall message, outcome, or conclusion, indicate ’yes’. If the statements contradict each other, present different information, or have distinct conclusions, indicate ’no’.
3. Compare the information presented in Statement 1 and Statement 2. Determine if the two statements convey the same or similar information, or if they contradict each other. Return ’yes’ if the statements match and ’no’ if they do not match.
4. Analyze the semantic meaning and factual content of two given statements and determine whether they convey the same information, describe the same event, or express the same idea, returning ’yes’ if they match and ’no’ if they do not match.
5. Do the two statements describe the same event or situation?

Table 5: Llama-generated prompts for claim matching.