Multi-LLM Verification for Question Answering under Conflicting Contexts

Geetanjali Rakshit and Jeffrey Flanigan
Computer Science and Engineering Department
UC Santa Cruz
{grakshit,jmflanig}@ucsc.edu

Abstract

Open-domain question answering (ODQA) often requires models to resolve conflicting evidence retrieved from diverse sources - a task that remains challenging even for state-of-the-art large language models (LLMs). While single-agent techniques such as self-verification and self-consistency have shown promise across natural language understanding and generation tasks, and multi-agent approaches involving collaborative or competitive strategies have recently emerged, their effectiveness for ODQA in the presence of conflicting contexts remains underexplored. In this work, we investigate these techniques using the QACC dataset as a case study. We find that incorporating a multi-agent verification step - where the best answer is selected from among outputs generated by different LLMs - leads to improved performance. Interestingly, we also observe that requiring explanations during the verification step does not always improve answer quality. Our experiments evaluate three strong LLMs (GPT-4o, Claude 4, and DeepSeek-R1) across a range of prompting and verification baselines.

1 Introduction

"Now produce your explanation and pray make it improbable."

- Oscar Wilde

Question answering (QA) has been a central area of research in natural language processing, with substantial progress over the past decade. Many models now perform near human-level accuracy on standard benchmarks (Rogers et al., 2023). As a result, the focus has increasingly shifted toward evaluating models under more challenging conditions and on more difficult datasets. State-of-the-art open domain question answering models typically retrieve information from a myriad of sources and look for the answer in them. These sources can contain conflicting evidence. At times, this can involve dealing with **ambiguity**, which may stem from the question itself - as seen in datasets like *SituatedQA* (Zhang and Choi, 2021) and *AmbigQA* (Min et al., 2020) - or from ambiguity and conflicts in the retrieved context, where different sources offer contradictory evidence (e.g., QACC (Liu et al., 2025), ConflictingQA (Wan et al., 2024)). The possibility of conflicting evidence arises due to reasons such as differing opinions and perspectives (Liu et al., 2021), presence of fake news and misinformation (Pan et al., 2021), information changing over time (Kasai et al., 2023) or presence of complementary information (Cattan et al., 2025). These scenarios pose significant difficulties even for powerful large language models (LLMs) (Liu et al., 2025). It is particularly important to correctly resolve contradictory evidence in answering questions in critical domains such as healthcare, because of misinformation present all over the internet, including data from unreliable sources.

Recent work has shown that prompting strategies designed to elicit explicit reasoning, such as **chain-of-thought prompting** (Wei et al., 2022) and **explanation generation** (Gu et al., 2023), can improve model performance on complex reasoning tasks. To further enhance factual correctness and answer reliability, several techniques have been proposed: **self-consistency** (Wang et al., 2022, 2024a), **self-refinement** (Madaan et al., 2023), and **chain-of-verification** (Dhuliawala et al., 2023). In parallel, a growing body of research explores **multi-agent frameworks**, including both collaborative and competitive strategies (Tran et al., 2025; Zhu et al., 2025; Luo et al., 2025; Adhikari and Lapata, 2025; Michael et al., 2023; Wang et al., 2024b; Chen et al., 2023).

Proceedings of Recent Advances in Natural Language Processing, pages 1012–1021
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-116

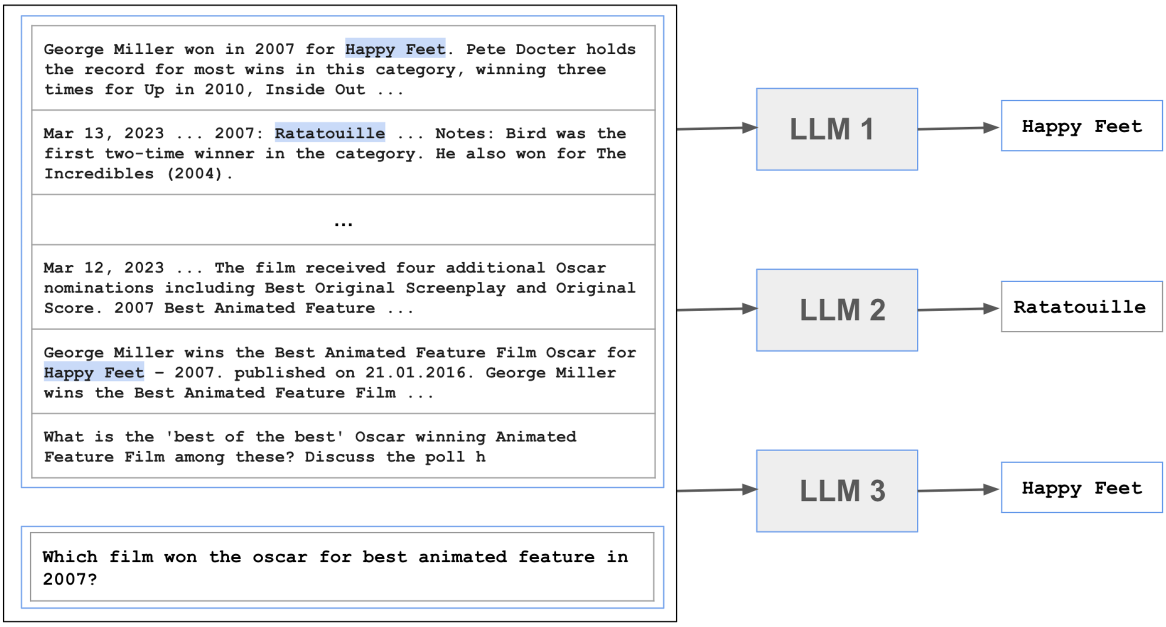

Figure 1: An example from the QACC dataset. It consists of a question and snippets from Google Search as contexts, which can contain conflicting information (highlighted in blue). It is challenging for LLMs to pick the correct answer from conflicting contexts.

In this work, we examine the effectiveness of these single-agent and multi-agent reasoning techniques for the specific challenge of answering questions from conflicting contexts. We focus our experiments on the QACC dataset (Liu et al., 2025), which is constructed from AmbigQA and contains approximately 25% of examples with conflicting evidence in the retrieved passages. Figure 1 illustrates an example from QACC.

Our observations can be summarized as the following:

- In a single-agent setup, DeepSeek-R1 achieves the best performance on QACC when prompted to produce both an answer and an explanation.

- Introducing a multi-agent verification step, where the best answer is selected from among independently generated LLM outputs, leads to improved performance over single-agent baselines. Further rounds of verification can offer additional gains.

- In multi-agent verification, using the same LLM for both answer generation and verification offers limited benefits. Incorporating model diversity leads to better performance.

Our code, prompts and data can be found at `https://github.com/geetanjali-rakshit/multiLLM-verification/`.

2 Experimental Setup

We evaluate both single-agent and multi-agent approaches to open-domain question answering (ODQA) under conflicting contexts using the QACC dataset (Liu et al., 2025). Our goal is to systematically examine how different prompting strategies and collaborative verification mechanisms affect performance, particularly in the presence of ambiguous or contradictory evidence. The dataset is described in section 2.1.

We use 3 state-of-the-art models in our experiments to benchmark performance on the QACC dataset: GPT-4o, Claude 4 and DeepSeek-R1. For the GPT-4o, we use the `gpt-4o-2024-11-20`. The Claude 4 model used is `claude-opus-4-20250514`. The DeepSeek-R1 model is `deepseek-chat`.

2.1 QACC Dataset

We use the QACC dataset, built on top of the AmbigQA dataset (Min et al., 2020), which augments unambiguous questions with context snippets retrieved via Google Search. Notably, 25% of the test examples include conflicting evidence

Model	Prompt	EM-C	EM-NC	EM-T	F1-C	F1-NC	F1-T
GPT-4o	No context	23.19	34.82	31.86	40.46	51.18	48.45
	Context	34.78	46.86	43.79	51.02	63.36	60.22
	Context first, then Answer	41.55	57.26	53.26	58.89	71.03	67.94
	Answer and Explain	**42.51**	**57.76**	**53.87**	**60.28**	**72.86**	**69.65**

Table 1: Performance comparison of different prompting strategies for GPT-4o. In row 1, no context is provided. In rows 2, 3 and 4, multiple contexts are provided. In row 3, the model is asked to pick a context first and then answer the question, while in row 4, the model is asked to generate the answer and then provide an explanation for the answer. (The numbers in **bold** represent the best results.)

across different retrieved passages. Each instance includes a question, multiple retrieved passages (contexts), and multiple plausible answers annotated with explanations.

We use the QACC development set for prompt engineering and tuning, and report results on the test set (813 examples). Evaluation metrics include:

- **EM-C / F1-C**: Exact Match and F1 on examples with *conflicting contexts*.
- **EM-NC / F1-NC**: Exact Match and F1 on examples with *non-conflicting contexts*.
- **EM-T / F1-T**: *Total* Exact Match and F1 across the full test set.

2.2 Single LLM Setup

We first benchmark performance in a single-agent setup using GPT-4o. We test several prompting strategies:

No context The model is asked to answer the question based solely on its internal knowledge or parametric memory, without being shown any contexts.

Context The model is asked to answer the question using all retrieved contexts.

Context first, then Answer The model is asked to select the most relevant context from all the retrieved contexts first, and then answer the question.

Answer and Explain The model is asked to answer the question using the contexts and then justify its answer with an explanation.

Findings

Our results using GPT-4o are summarized in Table 1. Prompting the model to generate an answer and explanation have the best overall results. In Table 2, we report these numbers for two other models: DeepSeek-R1 and Claude 4. Prompting the model to produce both an *answer* and *explanation* consistently yields the best performance (e.g., GPT-4o achieves EM-T = 53.87, F1-T = 69.65). Among the three LLMs, DeepSeek-R1 performs best overall in this single-agent setting.

In Table 4, we show an example of different answers and explanations generated by the LLMs. It is interesting to note that in *Answer1, Explanation1*, the LLM answers the question based on where the last name Tavarez *originates*, while in *Answer2, Explanation2*, the LLM answers the question based on where the last name Tavarez is most widespread. *Answer3, Explanation3* is focused on the most precise answer.

2.3 Multi Agent LLMs

We next explore a **multi-agent verification** framework, where an LLM is prompted to select the best answer from a set of candidate answers produced independently by multiple LLMs. We consider one round and two rounds of verification using each model in turn as a *verifier*. We prompt the *verifier* to choose the best answer from a list of 3 choices: the outputs from GPT-4o, Claude 4 and DeepSeek-R1, as *Round One* of verification. We do this with 3 different verifier LLMs. For *Round Two* of verification, we use these generated outputs to perform a second round of verification with each of the three LLMs.

2.3.1 Verification with Explanations

In this setup, the set of candidate answers as well as their corresponding explanations generated by the standalone LLMs is provided in the input to a verifier LLM. The verifier LLM selects the most appropriate answer-explanation pair. The prompt is in Table 7.

Setup	Model	EM-C	EM-NC	EM-T	F1-C	F1-NC	F1-T
Single LLM	GPT-4o	42.51	57.76	53.87	60.28	72.86	69.65
	DeepSeek-R1	44.44	60.4	56.33	62.16	73.76	70.81
	Claude 4	43.48	58.91	54.98	60.43	73.13	69.9
	Majority	44.44	60.4	56.33	62.16	73.76	70.81
Verifier - First Round	GPT-4o	44.93	59.9	56.09	62.83	74.3	71.38
	DeepSeek-R1	45.41	61.88	57.69	62.46	74.76	71.63
	Claude 4	**48.79**	62.87	59.29	**64.86**	74.6	72.12
	Majority	44.44	60.4	56.33	62.16	73.76	70.81
Verifier - Second Round	GPT-4o	45.41	61.55	57.44	63.15	**75.21**	72.14
	DeepSeek-R1	46.86	61.55	57.81	63.49	74.79	71.92
	Claude 4	48.31	**63.53**	**59.66**	63.8	**75.21**	**72.3**
	Majority	44.44	60.4	56.33	62.16	73.76	70.81
Previous SOTA (Liu et al., 2025)		47.34	57.26	54.74	59.61	69.79	67.19

Table 2: Performance comparison for using LLMs in a single agent versus multi-agent setup across models. The verification step chooses the best answer from among those generated by single LLMs. We report results from using GPT-4o, Claude 4 and DeepSeek-R1 as the verifier. We report results from performing two rounds of verification. (The numbers in **bold** represent the best results.)

Findings

Our results are summarized in Table 2. The main observations are the following:

- One round of verification improves EM-T and F1-T over all single-agent baselines.
- Two rounds of verification offer small additional gains.
- Claude 4 as the verifier performs best (EM-T = 59.66, F1-T = 72.3).
- However, majority-voting among generated answers underperforms compared to LLM-guided verification.

2.3.2 Verification without Explanations

To isolate the effect of explanations, we also evaluate a variant where the verifier selects from answer-only candidates.

Findings

Our results are summarized in Table 3. The main observations are the following:

- Verifying without explanations still improves over single-agent performance.
- Including explanations sometimes helps, particularly for F1 on conflicting examples, but is not uniformly beneficial.
- This suggests that explanations may introduce distractors or verbosity that harms answer selection in some cases.

In Table 4, we show an example of different answers and explanations generated by the single LLMs. It is interesting to note that the explanations shed light into what information is most convincing to the LLM to pinpoint the most accurate answer. While the explanations might not contribute much to improving the answer accuracy, but they are certainly rich in perspective.

2.3.3 Using the Same LLM Across Rounds

We test whether using the same model across multiple verification rounds helps. We use DeepSeek-R1 to generate upto three candidate answers, and then acts as its own verifier for two rounds.

Findings

Our results are summarized in Table 5. The main observations are the following:

- Verification still improves performance modestly.
- However, using different LLMs across verification stages consistently outperforms same-LLM pipelines, reinforcing the value of model diversity in collaborative settings.

Setup	Model	EM-C	EM-NC	EM-T	F1-C	F1-NC	F1-T
Verifier - First Round	GPT-4o	48.31	61.39	58.06	64.2	75.37	72.53
	DeepSeek-R1	47.83	62.71	58.92	63.52	75.02	72.09
	Claude 4	46.86	62.87	58.79	63.42	**75.49**	72.42
Verifier - Second Round	GPT-4o	47.34	62.71	58.79	63.98	75.35	72.45
	DeepSeek-R1	**48.31**	**63.2**	**59.41**	**64.49**	75.43	**72.65**
	Claude 4	46.86	62.71	58.67	63.1	74.93	71.92

Table 3: Performance comparison of verification rounds across models, without providing explanations. In this setup, we do not provide the explanations in the prompt, when we ask the LLM to chose the best answer from a set of answers. (The numbers in **bold** represent the best results.)

2.3.4 Majority Voting

In Table 2, we show the numbers for the performance metrics when we choose the majority answer from the set of generated answers (out of 3 in the case of single LLMs, 6 in the case of one round of verification, 9 in the case of two rounds of verification). In each case, the numbers are the same and worse than our best approach of single or two rounds of verifications. In Table 6, we show the average number of occurrences of the majority answer for each round.

3 Related Work

Recent research to improve results in a single-agent setup include **self-consistency**(Wang et al., 2022, 2024a), **self-refinement** (Madaan et al., 2023), and **chain-of-verification** (Dhuliawala et al., 2023), while multi-agent frameworks involve cooperative and competitive strategies.

Single Agent: Verification, Consistency, and Conflict Resolution

Self-consistency methods (Wang et al., 2022) have shown that sampling multiple reasoning paths and selecting the most consistent one improves factual accuracy. Extensions such as Soft Self-Consistency (Wang et al., 2024a) incorporate weighted voting based on explanation similarity. These approaches remain within the single-agent paradigm but share the spirit of aggregating diverse thoughts.

Verification chains (Chen et al., 2023) further enhance LLM reasoning by explicitly modeling a verifier agent that evaluates the correctness of candidate answers and explanations. Similarly, Chen et al. (2023) in RECONCILE introduce human-style correction demonstrations to teach models how to convince others or be persuaded, aligning answer revision with naturalistic argumentation.

Multi-Agent: Collaboration and Consensus

The RECONCILE framework (Chen et al., 2023) proposes a multi-model, multi-agent roundtable discussion mechanism where diverse LLMs share answers, confidence scores, and explanations to collaboratively refine and converge on a more accurate consensus. The framework demonstrates that incorporating confidence estimation and human-style rectification samples enables agents to revise their answers effectively. RECONCILE achieves substantial gains across seven reasoning benchmarks, outperforming GPT-4 on several tasks. Importantly, it emphasizes that model diversity (across different architectures and pretraining corpora) contributes significantly to the effectiveness of collaborative reasoning.

In contrast, Wang et al. (2024b) question the universal effectiveness of multi-agent discussions. They show that while discussion can be beneficial when prompts lack demonstrations, strong single-agent prompts can match or exceed the performance of multi-agent setups. Their CMD framework simulates structured group discussions and identifies cases where interaction leads to performance improvements, particularly when weaker LLMs interact with stronger ones.

Luo et al. (2025) address a specific failure mode in reasoning models termed the "Prefix Dominance Trap," where poor initial reasoning leads to persistent errors. Their LeaP framework enables reasoning paths to periodically share intermediate summaries with peers during inference, allowing for self-correction via cross-path insights. This peer-based collaborative mechanism leads to significant performance improvements on math and science

Question: Where does the last name Tavarez come from?
Contexts: *Context1:* This surname is most frequently occurring in The Dominican Republic, where it is held by 27,409 people, or 1 in 381. Apart from The Dominican Republic Tavarez ... *Context2:* 3385493 records ... Hispanic: variant of Portuguese and western Spanish Tavares. The surname Tavarez is found mainly in the Dominican Republic. Similar ... *Context3:* The surname Tavarez is found mainly in the Dominican Republic. Source: Dictionary of American Family Names 2nd edition, 2022. Similar surnames: Tavares, Labare, ... *Context4:* The most recent statistics we have for the Tavarez surname is from the 2010 census data. *Context5:* Early Origins of the Tavarez family. The surname Tavarez was first found in Castile, in north central Spain. *Context6:* Results 1 - 262 of 262 ... Search Tavarez family obituaries and memoriams on Legacy.com. There are 262 obituaries and memoriams for the surname Tavarez. *Context7:* The surname Tavarez is found mainly in the Dominican Republic. Source: DAFN2: Dictionary of American Family Names 2nd edition, ©2022 by Patrick Hanks and ... *Context8:* Tavares is a Portuguese surname. The Spanish version of this name is Tavárez. This surname was adopted by Sephardic Jews as well.
Answer 1: Portugal and Spain **Explanation 1:** The surname Tavarez is derived from the Portuguese surname Tavares, with the Spanish version being Tavárez, as mentioned in context 8. It is found mainly in the Dominican Republic but originates from these Iberian countries. **Answer 2:** The Dominican Republic **Explanation 2:** While the surname has origins in Portugal/Spain (as mentioned in contexts 2, 5, and 8), multiple contexts (1, 2, 3, and 7) explicitly state that the surname Tavarez is found mainly in the Dominican Republic, with context 1 providing specific statistics showing it's held by 27,409 people there. The contexts indicate this is where the surname is most prevalent today, making it the most direct answer to where the name comes from in terms of its current distribution. **Answer 3:** Spain **Explanation 3:** Context 5 explicitly states the origin of the surname Tavarez as being first found in Castile, in north central Spain, making it the most relevant answer for the question.

Table 4: Candidate answers and explanations generated by DeepSeek-R1, Claude 4 and GPT-4o. We observe that the LLMs are good at assimilating information from multiple sources to create a cohesive, convincing explanation.

Model	Setup	EM-C	EM-NC	EM-T	F1-C	F1-NC	F1-T
DeepSeek-R1	Single LLM (3 answer prompt)	44.44	**60.4**	56.33	**62.16**	**73.76**	**70.81**
	Verifier (Round 1)	46.38	**59.9**	56.46	61.16	72.04	69.27
	Verifier (Round 2)	**46.38**	59.41	56.09	61.51	71.93	69.28

Table 5: Performance comparison of single versus multi-agent setup using DeepSeek-R1 only. We prompt DeepSeek-R1 to generate up to 3 answers and then in the verification rounds 1 and 2, we prompt DeeepSeek-R1 to choose the best answer from the answer choices. (The numbers in **bold** represent the best results.)

Approach	count/choices
Round 0 (single LLM)	2.6/3
Round 1	4.9/6
Round 2	7.6/9

Table 6: Average count of majority answer in each round. Round 0 is the single LLM setup, at the end of which we take the majority over the set of answers generated by 3 LLMs (GPT-4o, Claude 4 and DeepSeek-R1). In Round 1 and Round 2 of verification, we take the majority out of the set of answers chosen (out of 6 choices in case of Round 1 and out of 9 choices in case of Round 2.

benchmarks, highlighting the importance of intermediate interaction rather than final answer aggregation alone.

MultiAgentBench (Zhu et al., 2025) introduces a benchmark suite for evaluating LLM-based agents in both cooperative and competitive environments. It offers a structured evaluation of task performance, coordination quality, and emergent social behavior across domains such as research, code generation, and planning. Notably, it finds that cognitive planning and graph-based communication protocols enhance milestone achievement and task success, establishing the utility of architectural and strategic diversity in collaborative multi-agent systems.

4 Conclusion

In this work, we explore a multi-agent collaborative framework for open-domain question answering from conflicting contexts, using the QACC dataset as a case study. Our approach leverages multiple large language models (LLMs) to generate diverse candidate answers, followed by a verification step - either with or without explanations - performed by a separate LLM agent. We find that this multi-agent verification mechanism consistently outperforms single-agent baselines, demonstrating the benefits of model diversity and structured answer selection in the presence of conflicting evidence.

Our results also suggest that while explanation generation can sometimes aid the verification process, it is not uniformly beneficial and may introduce noise depending on the task. Additionally, reusing the same model for both generation and verification yields marginal gains compared to setups that use different LLMs for these roles, highlighting the importance of heterogeneity in collaborative LLM systems.

5 Future Work

This study is limited to factoid questions in the QACC dataset. In future work, we plan to extend this framework to include non-factoid and open-ended questions, where reasoning, stance, and interpretation play a larger role. We are also interested in scaling our approach to include a larger pool of agents, investigating dynamic communication strategies, and exploring human-in-the-loop collaboration to further enhance reliability and trustworthiness in multi-agent LLM systems.

Limitations

In this study, we limit our experiments to the QACC dataset, which consists of factual questions and the answers are text spans from multiple contexts. We used the sources for each context in our prompts, but we didn't investigate the effect of the sources in selecting the answers. We also do not investigate answering yes/no questions that require reasoning over conflicting contexts (Nachshoni et al., 2025).

Instruction
You are provided with the following:
1. A set of contexts
2. A question
3. A list of possible answers, each tagged with an explanation.
Your task is to:
- Select the most concise and most befitting answer along with its explanation from among the provided list.
- If none of the answers in the list are satisfactory, you may provide a better answer and explanation. Make the answer as short as possible.
- You may choose an existing answer but provide an improved explanation, if necessary.
- Format your response as a single JSON object containing `"answer"` and `"explanation"` fields.

In-context example
Context1 from www.sho.com: Kerris Lilla Dorsey is best known for her roles as Brad Pitt's daughter in the Oscars nominated film MONEYBALL and as Steve Carell and Jennifer Garner's ...
Context2 from www.imdb.com: She is known for her roles as Paige Whedon in the television series Brothers & Sisters, Casey Beane, Billy Beane's (Brad Pitt) daughter, in the 2011 film ...
Context3 from people.com: Jul 2, 2015 ... Kerris Dorsey has worked with several of Hollywood's most famous dads. She currently plays Liev Schreiber's daughter on Ray Donovan and got ...
Context4 from www.imdb.com: She is known for her roles as Paige Whedon in the television series Brothers & Sisters, Casey Beane, Billy Beane's (Brad Pitt) daughter, in the 2011 film ...
Context5 from www.sportskeeda.com: Feb 11, 2023 ... All about Casey Beane, who was pictured in Moneyball as a young girl ... The 2011 film Moneyball was one of the most popular baseball movies ever ...
Context6 from en.wikipedia.org: Moneyball is a 2011 American biographical sports drama film directed by Bennett Miller with ... In the film, Beane (Brad Pitt) and assistant general manager Peter Brand ...
Context7 from www.latimes.com: Sep 10, 2011 ... And "Moneyball" is all about Brad Pitt. ... And while you get a couple of glimpses of Pitt's daughter, played by 13-year-old Kerris Dorsey, ...
Context8 from www.pinterest.com: Jan 9, 2020 ... (played by Brad Pitt) daughter, in the 2011 film Moneyball, ... Dorsey plays Bridget Donovan, the daughter of the title character, ...
Question: Who plays Brad Pitt's daughter in Moneyball?
Answer-Explanation pairs: [{ "answer": "Kerris Lilla Dorsey", "explanation": "This is the correct answer because it is explicitly stated in context 1 and 7. Unlike the other contexts, both of these contexts mention Brad Pitt and the movie Moneyball and the fact that Kerris Dorsey plays Brad Pitt's daughter, so we can confidently say that Kerris Lilla Dorsey is the most accurate answer." }, { "answer": "Casey Beane", "explanation": "This is the correct answer because Casey Beane is described as Billy Beane's (Brad Pitt's) daughter in context 2 and context 4" }, { "answer": "Bridget Donovan", "explanation": "This is the correct answer because Bridget Donovan is described as Brad Pitt's daughter in context 8." }]
Best Answer-Explanation: { "answer": "Kerris Lilla Dorsey", "explanation": "This is the correct answer because it is explicitly stated in context 1 and 7. Unlike the other contexts, both of these contexts mention Brad Pitt and the movie Moneyball and the fact that Kerris Dorsey plays Brad Pitt's daughter, so we can confidently say that Kerris Lilla Dorsey is the most accurate answer." }

Table 7: Our prompt (instruction and in-context example) for the verification step. All the text in black is part of the prompt, and the text in gray is where the completion would go in when the model is prompted.

References

Ashutosh Adhikari and Mirella Lapata. 2025. Debating for better reasoning: An unsupervised multimodal approach. *arXiv preprint arXiv:2505.14627*.

Arie Cattan, Alon Jacovi, Ori Ram, Jonathan Herzig, Roee Aharoni, Sasha Goldshtein, Eran Ofek, Idan Szpektor, and Avi Caciularu. 2025. Dragged into conflicts: Detecting and addressing conflicting sources in search-augmented llms. *arXiv preprint arXiv:2506.08500*.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Yuling Gu, Oyvind Tafjord, and Peter Clark. 2023. Digital socrates: Evaluating llms through explanation critiques. *arXiv preprint arXiv:2311.09613*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Siyi Liu, Sihao Chen, Xander Uyttendaele, and Dan Roth. 2021. Multioped: A corpus of multi-perspective news editorials. *arXiv preprint arXiv:2106.02725*.

Siyi Liu, Qiang Ning, Kishaloy Halder, Zheng Qi, Wei Xiao, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025. Open domain question answering with conflicting contexts. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1838–1854, Albuquerque, New Mexico. Association for Computational Linguistics.

Tongxu Luo, Wenyu Du, Jiaxi Bi, Stephen Chung, Zhengyang Tang, Hao Yang, Min Zhang, and Benyou Wang. 2025. Learning from peers in reasoning models. *arXiv preprint arXiv:2505.07787*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. 2023. Debate helps supervise unreliable experts. *arXiv preprint arXiv:2311.08702*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Eviatar Nachshoni, Arie Cattan, Shmuel Amar, Ori Shapira, and Ido Dagan. 2025. Consensus or conflict? fine-grained evaluation of conflicting answers in question-answering. *arXiv preprint arXiv:2508.12355*.

Liangming Pan, Wenhu Chen, Min-Yen Kan, and William Yang Wang. 2021. Attacking open-domain question answering by injecting misinformation. *arXiv preprint arXiv:2110.07803*.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What evidence do language models find convincing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7468–7484.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024a. Soft self-consistency improves language models agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–301.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024b. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *arXiv preprint arXiv:2402.18272*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Michael JQ Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. *arXiv preprint arXiv:2109.06157*.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. 2025. Multiagentbench: Evaluating the collaboration and competition of llm agents. *arXiv preprint arXiv:2503.01935*.