Seemingly Plausible Distractors in Multi-Hop Reasoning:Large Language Models Attentive Readers?

Neeladri Bhuiya Viktor Schlegel Stefan Winkler

Abstract

State-of-the-art Large Language Models (LLMs) are accredited with an increasing number of different capabilities, ranging from reading comprehension over advanced mathematical and reasoning skills to possessing scientiflc knowledge. In this paper we focus on multi-hop reasoning—the ability to identify and integrate information from multiple textual sources. Given the concerns with the presence of simplifying cues in existing multi-hop reasoning benchmarks, which allow models to circumvent the reasoning requirement, we set out to investigate whether LLMs are prone to exploiting such simplifying cues. We flnd evidence that they indeed circumvent the requirement to pedorm multi-hop reasoning, but they do so in more subtle ways than what was reported about their fine-tuned pre-trained language model (PLM) predecessors. We propose a challenging multi-hop reasoning benchmark by generating seemingly plausible multi-hop reasoning chains that ultimately lead to incorrect answers. We evaluate multiple open and proprietary state-of-the-art LLMs and show that their multihop reasoning performance is affected, as indicated by up to 45Vo relative decrease in Fl score when presented with such seemingly plausible alternatives. We also find that—while LLMs tend to ignore misleading lexical cues—misleading reasoning paths indeed present a significant challenge. The code and data are made available at https://github.com/zawedcvg/Are-Large-Language-Models-Attentive-Readers.

1 Introduction

Recent developments in the field of language modelling and the introduction of open (Touvron et al., 2023) and proprietary (OpenAI, 2023) Large Language Models (LLMs) have undeniably advanced the state of the art in Natural Language Processing (NLP). LLMs have been credited with various understanding and reasoning capabilities, ranging from arithmetic (Cobbe et al., 2021), deductive (Saparov et al., 2023) and formal (Schlegel et al., 2022b; Madusanka et al., 2023) reasoning and possessing general (AlKhamissi et al., 2022), and domain-specific (He et al., 2023) knowledge. Due to their size and generalisation capabilities (Brown et al., 2020), their evaluation on benchmarks requiring such types of reasoning is typically performed in zero- or few-shot settings on many NLP tasks, without the need for fine-tuning datasets.

These zero- and few-shot capabilities seem to alleviate one of the weaknesses identified with the previous generation of fine-tuning based NLP architectures such as transformer-based (Vaswani et al., 2017), and pre-trained language models (Devlin et al., 2019)—the reliance on data-set specific 'artefacts" (Gururangan et al., 2018; Schlegel et al., 2022a) and, as a consequence, lack of generalisation beyond specific datasets. For example, in one of the popular reading comprehension and reasoning benchmarks (Dua et al., 2019), the majority of questions starting with 'How many" can be answered correctly with "2". Following standard fine-tuning practice and splitting data in train and test randomly, such a simple heuristic will be present in both training and evaluation data, so a fine-tuned model will learn it and obtain high scores, without necessarily performing reasoning.

LLMs seemingly circumvent this issue, as they are not fine-tuned on benchmark data. As such, they are not exposed to simplifying dataset artefacts by design, and it is reasonable to assume that they do not learn to exploit them.

However, while there is a growing body of work investigating the strengths and limitations of LLMs (Huang et al., 2023b), little research has been carried out to validate this assumption, and to investigate whether and to what extent LLMs inherit the 'dataset artefact" weaknesses of their fine-tuned predecessors. This is an important research question to pursue, motivated by recent findings on benchmark leakage into pre-training or instruction-tuning data (Deng et al., 2024), which invalidate the zero-shot setting and potentially allow LLMs to learn such dataset artefacts. Another line of research suggests that LLMs tend to 'over-reason" (Chiang and Lee, 2024), perhaps due to "sycophancy" (Perez et al., 2023), i.e., the tendency to generate the presumably preferred answer over the correct one, leading to complicated reasoning where none is required.

In this paper, we turn our attention to the well-studied capability to perform multi-hop reasoning and reading comprehension—that is, to integrate textual information from multiple different source documents. Typically, this capability is evaluated by asking questions where the necessary information to arrive at the correct answer is spread across multiple documents (Yang et al., 2018a; Welbl et al., 2018; Inoue et al., 2020). It is important to understand to what extent NLP methods possess this capability, as it is required for many real-world tasks, such as retrieval-augmented generation (Lewis et al., 2020) when summarising retrieved documents, and because it is a necessary prerequisite to human-level reading comprehension (Kintsch, 1988).

Previous work has shown that NLP architectures might possess inadequate capabilities to perform multi-hop reasoning (Min et al., 2019a). However, these findings were established before the advent of large language models. To have a clear understanding of the limitations of the capabilities of state-of-the-art research, it is crucial to re-investigate these claims with the current generation of LLM-based approaches (Bowman, 2022). While there is vivid research on (open-book) multi-hop reasoning capabilities of LLMs (Sakarvadia et al., 2023; Liu et al., 2023; Yang et al., 2024), how well they perform when presented with multiple, seemingly plausible multi-hop reasoning paths remains unclear.

To address this gap, we focus on the capability of LLMs to perform multi-hop reasoning when multiple seemingly plausible answers are present, where only minor details invalidate the alternative. We show that existing methods—calibrated to evaluate pre-LLM architectures—are inadequate to evaluate LLMs, and that LLM reasoning failures are indeed distinct from their fine-tuned PLM predecessors. We present a methodology to generate challenging examples with 'plausible distractors" to evaluate LLMs' capabilities to perform multi-hop reasoning when presented with seemingly correct, but ultimately wrong and thus distracting evidence. Our results show that the reasoning capabilities of a range of open and proprietary LLMs, including GPT-4, are affected by these 'plausible distractors".

2 Related Work

It has been shown that basic pattern matching (Schlegel et al., 2020) and one-hop (Min et al., 2019a) models can solve a large proportion of questions in multi-hop question answering datasets, presumably because the answer sentence often contains keywords common with the question, thus negating the need to follow a reasoning path and attend to multiple documents. Particularly HotpotQA (Yang et al., 2018b), due to its multi-hop question design, was the subject of multiple studies. Approaches architecturally incapable of multi-hop reasoning still achieved close to state-of-the-art performance (Min et al., 2019a; Trivedi et al., 2020), suggesting questions answerable in such a way do not necessitate multi-hop reasoning.

In light of these results, several adversarial attacks have been proposed to check whether the dataset evaluates multi-hop reasoning without exhibiting "shortcuts", by ensuring that the correct answer can only be procured if the evaluated model can retrieve and combine information from distinct reasoning hops. Jiang and Bansal (2019) elicited distracting paragraphs by using the titles of the gold paragraphs and the answer, which are subjected to phrase-level perturbations and word replacement, thus creating a distracting paragraph. Others decomposed the multi-hop questions in multiple single questions (Min et al., 2019b; Perez et al., 2020; Ding et al., 2021) (e.g. DecompRC in Figure 2) showed that the—typically BERT- or other PLM-based—fine-tuned SOTA models struggled to answer both sub-questions correctly when answering the complete question, or were distracted by their alterations, suggesting the presence of reasoning shortcuts (Tang et al., 2021).

By design, these methods bear only negative predictive power (Gardner et al., 2020): failing to see a performance drop does not imply that the model performs the evaluated capability well, but rather that the methodology might have limited suitability to evaluate the investigated phenomenon, i.e., multi-hop reasoning. As the methodologies presented above focus on fine-tuned models, they assume that multi-hop reasoning is circumvented through simple, lexical similarity-based methods like word matching. For example, Jiang and Bansal (2019) do not consider that their generated paragraphs are isolated, as they contain no explicit reference to other paragraphs in the context, such as a shared named entity. Meanwhile, Ding et al. (2021) only add a single distracting sentence. Thus, simple word matching, which ensures that the final answer is of the same entity type as in the question, can often lead to the correct answer. This might not be sufficient for LLMs, as they—due to their size and emergent capabilities—might circumvent multi-hop reasoning by exploiting more subtle textual cues. Indeed, in our empirical study, we show that existing methods, due to these limitations, do not adequately test an LLM's reasoning capabilities.

Therefore, to analyse an LLM's ability to reason more adequately, we go beyond the state of the art and introduce a novel method to more effectively evaluate the multi-hop reasoning capabilities of LLMs. Specifically, we ensure the continuity of seemingly plausible alternative reasoning paths, which lead to answers that are ultimately wrong. To succeed, the model is required to pay close attention to small yet important details in the questions and paragraphs.

This ability is important practically, for example when an LLM is prompted to evaluate/summarise the outcome of a debate, where both sides will present plausible arguments with only one being ultimately correct (Sun et al., 2023; Li et al., 2024). With LLMs increasingly used to judge and improve (other) LLMs' potentially similar outputs on the same topic (Huang et al., 2023a), it is important to establish, if they possess the necessary prerequisites to do so. More broadly, similar to other works in this line of research, we look at linguistic competence rather than performance (Chomsky, 1965): if we accredit multi-hop reasoning capabilities to LLMs, then, similar to humans, we expect them to exhibit these capacities not only in the majority of cases but in edge case scenarios as well, such as when presented with seemingly plausible alternate reasoning paths.

In this section, we describe our approach to evaluating the multi-hop reasoning capabilities of LLMs. We do so by creating "distractor" paragraphs that present seemingly plausible yet incorrect alternative paths in the reasoning chain while ensuring that this process doesn't affect the final solution.

First, the question is treated as a two-hop question and converted into two sub-questions. This is done to be able to branch out alternative reasoning paths from each of the sub-questions. The sub-questions are analyzed to identify modifiable portions, which are then manipulated to create 'distractor" sub-questions that lead to a different answer and thus a different reasoning chain, which is ultimately wrong, as the models are presented with the original, unmodified question. The 'distractor sub-questions" are finally used to generate 'distractor paragraphs" containing 'distractor answers” utilizing an LLM.

The method comprises three main steps: I. Acquiring the main entity, II. Extracting its modifiable details, and III. Creating the distractor paragraphs.

I. Acquiring the main entity We use the human-annotated sub-questions from Tang et al. (2021), as exemplified in Figure 2. We define main entities as those that are the focus of the question. For example, in Figure 2, the main entities for the sub-questions would be ‘movie stars” and ‘year” respectively. We choose the ‘main entity” in each sub-question, using a few dependency parse-based rules[1]. Intuitively, we exploit the relations between the “wh”-word and other noun phrases to extract the main entity. Specifically:

(i) If the “wh” question word [ILLEGIBLE] is the root, and there exists a word A with a dependency nsubj or nsubj:pass with WH as the head, A is the main entity.

(ii) Alternatively, if there exists a word A with a dependency of type det, nsubj, or nsubj:pass with a wh-word [ILLEGIBLE]:

(a) If A is a noun, A is the main entity.

(b) Otherwise, if A is a verb, the word B having a relation acl:relcl with B being the head, we mark B as the main entity.

(iii) Else, if any word A has a dependency with a word B of type nsubj or nsubj:pass, where B is the word with a direct dependency with the wh-word, A is assigned as the main object.

II. Extracting the details Next, we extract the details that need to be manipulated to create the distractor question. The main idea is to obtain modifiers of any entity in the question other than the main entity (from the previous step). Specifically:

(i) For any dependency between two words C and D, we check if the dependency is of the form obl, obj, nsubj, or nsubj:pass. We also ensure that D isn’t the main entity identified in the previous step.

(ii) If the above rule is satisfied, we check if C or D has a dependency appos with any named entity.

(iii) If there is no such relation, modifiers of D of the form nummod, amod, nmod, compound, or flat are used to get modifiable parts if the modifier isn’t the main entity identified in the previous steps.

We extract the modifiers and not the object they modify for two reasons: First, changing the object often causes the overall question to become nonsensical. Secondly, changing the modifier ensures a minimal yet semantically meaningful modification of the question (Schlegel et al., 2021).

III. Creating the distractor paragraphs After obtaining modifiable parts, we distinguish whether these are Named Entities or not. For each of the named entities, we obtain their type using [ILLEGIBLE] et al. (2020)’s Named Entity Recognition (NER) processor. We then generate a fake entity of the same type with the help of GPT-4.

Next, for the non-named entities, we use RoBERTa’s (Liu et al., 2019) masked token prediction objective to obtain alternative words. Specifically, we mask the modifiable parts and sample the top ten probable tokens from the language model. To ensure that the new word is sufficiently different yet still plausible given the context, we establish the following constraints empirically:

- Sentence Similarity of the new sequence in comparison to the initial question, as given by the cosine similarity of all-mpnet-base-v2 (Reimers and Gurevych, 2019) is < 0.991;
- Word similarity under RoBERTa of the original word and the word replacing it is < 0.4;

[1] By ‘Dependency of type C between A and B” we mean A is the head, B is the dependent and C is the relation type. See Appendix for type definitions.

- Perplexity, i.e. the RoBERTa predicted probability of the new sentence, is > 0.001.

The new words and named entities are used to create new fake questions. We use these fake questions to create fake question tuples, i.e., fake questions for the different hops. While generating the fake question tuples, we mask the tokens in the second sub-question corresponding to the first sub-question's answer. Next, we feed these fake tuples into GPT-4 and ask it to generate the distractor paragraphs. We generate a pair of distractor paragraphs for each tuple. Figure 3 shows the instantiation of our proposed method on a single example, with the generated distractor paragraphs and the corresponding gold paragraphs. In the attack each of these distractor paragraphs replaces one of the non-gold paragraphs, to prevent adding extra tokens and to ensure that the ratio of 2 gold paragraphs and 8 distractor paragraphs of the distractor setting of HotpotQA is maintained.

Data Quality Following this procedure, we generate 132 instances of the 'other" type, while 547 are created from named entities. To ensure that the generated distractor paragraphs are valid, do not contradict the gold paragraphs, and do not cause contradictions with the label, we randomly sample and inspect 100 named entity-based and all 132 of the 'other" examples. For the former, none of the sampled examples were contradictory. For the latter, 13 were found to have either one or both of the distractor paragraphs contradictory—those examples were discarded. Furthermore, we conducted a user study (see Appendix F), which showed that humans have no difficulty extracting the correct answer when given a combination of real and distractor paragraphs. It was also reported that the distractor paragraphs seldom contain contradicting information. We further compare the word count of the adversarial and the original paragraphs to check if the adversarial paragraphs artificially increase complexity through a larger word count. On average, the adversarial paragraphs had a word count of 81.2, slightly lower than the average word count of the original paragraphs, which is 95.95.

Through manual verification, a user study, and the comparison of the word count of plausible paragraphs and their counterpart real paragraphs, we can conclude with high certainty that the plausible paragraphs don't contain contradictory information, and that the drop in performance of the models is due to their inherent weakness and not some artificially added complexity.

3 Experiment Setup

First, we investigate LLM's capabilities and limitations compared to previous PLM-based state of the art. Then, we evaluate the multi-hop reasoning capabilities of LLMs using our proposed methodology. Finally, we conduct an in-depth analysis of what makes reasoning hard for LLMs on our benchmark and conclude by evaluating state-of-the-art LLMs and prompting techniques. Unless mentioned otherwise, we use the chat models for Llama-2.

Do LLMs suffer from the same flaws as fine-tuned models? Llama-2-13B (Touvron et al., 2023) is used as the baseline LLM. We evaluate using few-shot prompts, as these allow the model to stick to the expected output format better than zero-shot. This setting is used throughout the paper unless mentioned otherwise. Two styles of prompts were used, normal and chain of thought, as per the strategies discussed in Wei et al. (2023). All reported metrics are measured at token level and averaged across all the instances, following standard evaluation practice (Yang et al., 2018b).

We test the LLMs' performance when attacked with AddDoc (Jiang and Bansal, 2019), an adversarial attack on HotpotQA for BERT-based models. This is intended to check an LLM's ability to handle "distracting" paragraphs. SubQA was used to determine if the models could answer the individual questions before answering the entire question. It is a sample of 1000 questions and their sub-questions from the dev set of HotpotQA, with the sub-questions being human-verified. This allows us to evaluate model consistency in answering both the multi-hop question as well as the individual sub-questions correctly. It also allows us to investigate the opposite: When the (more complex) composite question is answered correctly, but either of the (simpler) decomposed questions is answered wrongly, the model might rely on some reasoning shortcuts, discarding sub-question information. Finally, we evaluate if LLMs can retrieve the correct answer when necessary information from one of the gold paragraphs is missing, using the DiRe test set (Trivedi et al., 2020).

Do LLMs get distracted by seemingly plausible alternate reasoning paths? As described in Section 3, the attack aims to create paragraphs that provide irrelevant information that is closely related to the property/entity being questioned about. Here, we evaluate a representative sample of open-source and proprietary LLMs, specifically, Llama-2-13B, Llama-2-70B, Mixtral-8x7B-Instruct-v0.1, GPT-3.5 and GPT-4. To contextualise the performance of LLMs to their fine-tuned PLM counterparts, we also fine-tune a longformer model on the HotpotQA training set and evaluate it on our proposed benchmark (see Appendix for details). Based on the chatbot leaderboard (Chiang et al., 2024) at the time of writing, the best state-of-the-art model was GPT-4. Thus we evaluate GPT-4 to investigate how our findings generalise to stronger models.

What are the effects of the different parameters? Experiments are conducted to check the impact of the method's parameters on the performance of LLMs. Specifically, the different parameters we investigate are: 1) number of "distractor" paragraphs generated, i.e., two or four; 2) whether the distractor paragraphs are generated from the two sub-questions belonging to the same multi-hop question or if the sub-questions belong to two independent multi-hop questions; 3) The type of modifiable portion that is changed in the sub-question, i.e., Named Entity or not; 4) whether the paragraphs, if not generated from two distinct sub-questions, are both generated from the second sub-question.

In this section, we present the results of our experiments, compare them against prior work, and discuss deeper insights. Unless otherwise stated, all reported results of the adversarial attack are statistically significant at $p < 0.05$, determined by conducting a one-sided Student's t-test.

5.1 Do LLMs suffer from the same flaws as fine-tuned models?

I. Setting up the baseline Llama-2-13B chat model is used as the baseline for the performance of an LLM in a zero/few-shot setting; results are shown in Table 1. The F1 score indicates that the few-shot setting without chain-of-thought prompting performs best. This is because in the chain of thought setting the model often gives a lengthy explanation, thus reducing precision and F1 score.

II. Reasoning shortcuts using SubQA Table 2 shows the result of running few-shot Llama-2-13B in the controlled setting on the SubQA dataset. Llama-2 performs much better on the individual sub-questions than the question requiring multi-hops. This finding, in line with analyses focusing on fine-tuned models (Tang et al., 2021), suggests some inconsistencies in its reasoning capabilities and difficulty in combining information from multiple sources.

Table 3 indicates the performance statistics for individual samples. F1 > 0.5 is used here to evaluate a question as correct. The first row consists of questions where the individual sub-questions and the whole question were answered correctly. The second row indicates the questions where the final answer was correct despite getting the individual hops wrong, while the third is where the final answer was incorrect despite the individual hops being correct. 10

III. Reasoning shortcuts in DiRe DiRe consists of removing the bridging gold paragraph from the context, with the claim that a model should not be able to answer them under these conditions, and if they are, the examples exhibit a reasoning shortcut exploited by the model. Table 5 shows the results of Llama-2-13B on this. Surprisingly, the model still maintains a decent performance level, confirming that HotpotQA indeed contains several reasoning shortcuts. Seemingly, LLMs—similar to their fine-tuned predecessors—readily exploit such shortcuts despite not being explicitly trained on HotpotQA.

IV. Reasoning failures when presented with distracting paragraphs from AddDoc Table 6 shows the performance of Llama-2-13B, Llama-2-70B and Mixtral-8x7B-Instruct-v0.1, in few-shot prompt setting, when attacked with the first 2000 examples of AddDoc (Jiang and Bansal, 2019), the most successful method to show reasoning weaknesses of models fine-tuned on HotpotQA, by adding crafted paragraphs which are lexically similar to the question. Apparently, and in stark contrast to fine-tuned models, LLMs performance does not drop on the benchmark, even slightly increasing for some of the evaluated models. This finding suggests that the reasoning shortcuts exploited by LLMs are indeed less obvious than simple lexical overlap, thus further motivating the need for a more sophisticated method to evaluate multi-hop reasoning, such as those proposed in this paper.

5.2 Do LLMs get distracted when faced with seemingly plausible alternatives?

Table 4 shows the results of various open- and closed-source LLMs using our proposed benchmarking method. All models show a significant drop in their F1 scores and their Exact-Match (EM) scores. Importantly, this seems to be a model property rather than an artefact of the prompting technique, as the behaviour persists across different prompting methods (see Appendix H). Furthermore, even GPT-4 exhibits a drop of 14 points in F1 under the strongest adversarial attack setting i.e., when adding four adversarial paragraphs (see Appendix G). This is remarkable, as the benchmark was partially generated with GPT-4 in the loop. This highlights the feasibility of our method to evaluate a model using an equally strong model as an adversary, a property that other benchmarks tend to lack (Zellers et al., 2018, 2019).

5.3 Analysing the effects of different parameters

Next, we investigate which settings contribute most to the drop in performance.

Count of distractor paragraphs As we can modify the number of alternate reasoning chains, and thus generate distractor paragraphs, it is worthwhile investigating whether increasing their number leads to decreased performance. Table 4, "Paragraph count" columns, shows the results of the various models in the chain of thought few-shot setting when facing two or four distractor paragraphs, respectively. Indeed, the higher the number of adversarial paragraphs, the more the model struggles, with an additional decrease of about 10 F1 points for every fake reasoning chain on average.

Are the paragraphs related? As our method creates fake sub-questions that are used to generate distractor paragraphs, we can modify if the paragraphs to be used in the attack belong to the same fake question pair or not. If not, the attack will use paragraphs from different pairs but will ensure that if k adversarial paragraphs are being added, $k/2$ are generated from the first sub-question and the other from the second sub-question. This is useful to check if models struggle because of the presence of alternate multi-hop reasoning chains, or if the difference in performance is attributed to distractor paragraphs containing similar but otherwise unrelated information.

Table 4, columns "Paragraph Related" shows the performance of the models in this setting. For Llama-2-13B, Mixtral-8x7B-Instruct-v0.1, and Llama-2-70B, related paragraphs, and therefore complete alternate reasoning chains, cause a larger drop than unrelated distractor paragraphs. Interestingly, GPT-3.5 exhibits the opposite behaviour, performing slightly worse when an alternate reasoning chain does not connect the distractor paragraphs.

Modified type Because the main entity of the question can be either part of a Named Entity or not, we can distinguish model performance between these settings, Table 4, columns "Modified Type", shows the results of this test. Aside from Llama-2-13B, which performs significantly worse on Named Entities, the differences are not statistically significant, indicating that both distractor types seem to be equally difficult.

Are the paragraphs unrelated and only belong to the 2nd subquestion? We have shown that (with the exception of GPT-3.5) examples containing fake paragraphs related by a seemingly alternate reasoning chain are harder for LLMs to process correctly. Similarly, we can investigate if fake paragraphs that are generated purely from the second sub-question add further complexity. Since the paragraph generated from the second sub-question is the only paragraph that contains an entity of the same type as the actual answer, the rationale is to investigate what contributes more to hard multi-hop reasoning: producing seemingly alternate reasoning chains or just adding adversarial paragraphs similar to the paragraph answering the second sub-question. We ensure that the number of adversarial paragraphs, generated using our method, is the same in both settings.

As can be seen in the last column of Table 4, "Second Sub-Q only", all LLMs perform worse when the paragraphs are not generated from the second sub-question only, thus adding further evidence to the hypothesis that examples with seemingly plausible alternate reasoning chains are indeed harder for LLMs to process correctly. Additionally, only the fine-tuned longformer model exhibits the opposite behaviour, suggesting that PLM-based fine-tuned models indeed tend to learn more simple word-matching type heuristics, as generating multiple paragraphs from the second sub-question results in more fake paragraphs that are lexically similar to the question and answer sentence. This adds further evidence that there is a need to reevaluate the weaknesses of LLMs, as insights derived from PLMs do not necessarily carry over.

The second sub-question-only setting is most similar to AddDoc (Jiang and Bansal, 2019) and other existing attacks on HotpotQA. However, unlike for AddDoc, all LLMs still show a drop in performance. This demonstrates the effectiveness of generating adversarial paragraphs by changing minute details extracted from the question, surpassing the impact of existing attacks. The paragraphs generated in this manner challenge the LLMs more effectively, highlighting their susceptibility to being "blinded by nuance".

4 Conclusion

We explored whether LLMs can perform multi-hop reasoning when presented with seemingly plausible alternative reasoning paths. We showed that, while LLMs are less susceptible to simple lexical overlap attacks than fine-tuned PLM-based models, they are still significantly affected by more subtle distractors that form coherent but ultimately incorrect reasoning chains.

We introduced a methodology to construct such plausible distractor paragraphs by minimally modifying decomposed sub-questions and generating alternative reasoning paths. Using this method, we built a challenging benchmark and evaluated a range of open and proprietary LLMs. The results show consistent and statistically significant drops in performance across models, including GPT-4, indicating that seemingly plausible but incorrect reasoning paths pose a substantial challenge.

Our findings suggest that existing evaluation methods are insufficient to fully assess multi-hop reasoning in LLMs and that stronger, reasoning-aware adversarial benchmarks are necessary. We hope that our proposed methodology and benchmark contribute to more robust evaluation and future improvements in LLM reasoning capabilities.

Limitations

[LIMITATIONS SECTION CONTENT NOT FOUND IN AVAILABLE PARSED PDF TEXT — MISSING]

References

[REFERENCES SECTION CONTENT NOT FULLY AVAILABLE IN PARSED PDF TEXT — MISSING]

[APPENDIX SECTION CONTENT NOT AVAILABLE IN PARSED PDF TEXT — MISSING]