# Compare Results

Old File:

**2024.emnlp-main.147.pdf**

**15 pages (15.70 MB)**

versus

New File:

**2024_emnlp-main_147.pdf**

**28 pages (177 KB)**

2/8/2026 5:27:34 AM

**Total Changes**

50

**Content**

9 Replacements

27 Insertions

14 Deletions

**Styling and Annotations**

0 Styling

0 Annotations

Go to First Change (page 1)

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

Neeladri Bhuiya[1,2], Viktor Schlegel[*3,4], and Stefan Winkler[1,2]

[1]ASUS Intelligent Cloud Services (AICS) Singapore
[2]National University of Singapore
[3]University of Manchester, United Kingdom
[4]Imperial Global Singapore
[*]Corresponding author: neeladri.bhuiya@u.nus.edu, v.schlegel@imperial.ac.uk, winkler@nus.edu.sg

**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 scientific 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 find evidence that they indeed circumvent the requirement to perform 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 multi-hop reasoning performance is affected, as indicated by up to 45% relative decrease in F1 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-R

[*]Work done while author was at AICS.

# 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 dataset specific "artifacts" [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 artifacts 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 artifact" 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 artifacts. 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 infor-

mation 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 rea-

soning.

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. Jiang2019 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 1) 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, Jiang2019 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, Ding2021 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

Original Q: What year did Guns N' Roses promote a movie starring Arnold Schwarzenegger as a former New York Police detective?
Sub-question 1: Which movie stars Arnold Schwarzenegger as a former New York Police detective?
Sub-question 2: What year did Guns N' Roses perform a promo for (answer from sub-question 1)?

Figure 1: Example of a decomposed multi-hop question.

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.

# 3 Methodology

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.

## 3.1 I. Acquiring the main entity

We use the human-annotated sub-questions from Tang2021, as exemplified in Figure 1. We define main entities as those that are the focus of the question. For example, in Figure 1, the main entities for the sub-questions would be "movie" and "year" respectively. We choose the "main entity" in each sub-question, using

a few dependency parse-based rules. Intuitively, we exploit the relations between the "wh"-word and other noun phrases to extract the main entity. Specifically:

1. If the "wh" question word $WH$ 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.

2. Alternatively, if there exists a word $A$ with a dependency of type `det`, `nsubj`, or `nsubj:pass` with a wh-word $WH$:

   (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.

3. 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.

### 3.2 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:

1. 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.

2. If the above rule is satisfied, we check if $C$ or $D$ has a dependency `appos` with any named entity.

3. 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)].

Original Q: The arena where the Lewiston Maineiacs played their home games can seat how many people?
Sub-Q1: Which arena did the Lewiston Maineiacs play their home games?
Sub-Q2: How many people can the Androscoggin Bank Colisee seat?
Fake paragraph 1: The Lewiston Maineiacs took to the ice at the Maple Leaf Arena for their thrilling playoff games.
Fake paragraph 2: Maple Leaf Arena, known for its state-of-the-art facilities and spacious seating, can accommodate an impressive number of 15,000 spectators.
Gold Paragraph 2: The Lewiston Maineiacs [...] played its home games at the Androscoggin Bank Colisee. [...]

Figure 2: Instantiation of our proposed method. With "arena" as main entity of sub-question 1, we extract "home" to be replaced with "playoff". Then, we use the modified sequence with the original sub-question 2 (masking the answer "Androscoggin Bank Colisee") as prompt to GPT-4 to generate the distractor paragraphs 1 and 2. The distractor paragraphs generated have "Maple Leaf Arena" as the bridging entity in the false reasoning chain which leads to the wrong answer "15,000".

### 3.3 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 Qi2020'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 $< 0.4$;
- 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 2 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.

### 3.4 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.

## 4 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.

### 4.1 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 Wei2023. 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)].

### 4.2 Do LLMs get distracted by seemingly plausible alternate reasoning paths?

As described in Section **??**, 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 and Lee(2024)] at the time of writing, the best state-of-the-art model was GPT-4. Thus we evaluate GPT-4 to investigate

Table 1: Comparing normal and chain-of-thought prompts using Llama-2-13B as baseline

| Type | F1 score | Precision | Recall | Ans# words |
|---|---|---|---|---|
| Few-shot | 0.500 | 0.488 | 0.599 | 3.08 |
| CoT | 0.479 | 0.468 | 0.599 | 5.08 |

how our findings generalise to stronger models.

### 4.3 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.

## 5 Experiment Results

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?

#### 5.1.1 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.

Table 2: Results of Llama-2-13B on SubQA dataset

| Type | F1 score | Precision | Recall |
|---|---|---|---|
| Sub-question 1 | 0.743 | 0.161 | 0.789 |
| Sub-question 2 | 0.693 | 0.691 | 0.782 |

Table 3: Breakdown of the results on running SubQA

| Type | Accuracy |
|---|---|
| All correct | 0.414 |
| Correct but sub-questions wrong | 0.107 |
| Wrong but both sub correct | 0.250 |

#### 5.1.2 II. Reasoning shortcuts using SubQA

Table 2 shows the result of running few-shot Llama-2-13B 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. 10.7% of the questions were answered correctly without getting both sub-questions correct. This accounts for over 20% of the questions that the model got correct, which is considered model failure by Tang2021, thus indicating that the model indeed follows some form of shortcuts in its multi-hop reasoning process. However, this percentage is much lower than for PLM-based fine-tuned models, which reach close to 50% [Tang et al.(2021)]. For 25% of the questions, the model got both sub-questions correct but was unable to combine them to give the final answer, thus demonstrating difficulties in bridging and integrating separate information during multiple reasoning hops.

Table 4: Llama-2-13B performance on DiRe when using a normal (non-CoT) prompt and priming with few-shot examples.

| Dataset | F1 score |
|---|---|
| Original dataset of 4174 examples | 68.7 |
| DiRe preprocessed | 50.4 |

Table 5: F1 score of Llama-2-13B, Llama-2-70B and Mixtral-8x7B-Instruct-v0.1 when attacked with 2000 examples of AddDoc in the few-shot setting.

| Model | Original | AddDoc |
|---|---|---|
| Llama-2-13B | 50.3 | 51.7 |
| Mixtral-8x7B-Instruct-v0.1 | 58.0 | 58.0 |
| Llama-2-70B | 53.9 | 54.6 |

#### 5.1.3 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 4 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.

#### 5.1.4 IV. Reasoning failures when presented with distracting paragraphs from AddDoc

Table 5 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.

Table 6: Results of Llama-2-13B, Mixtral-8x7B-Instruct-v0.1, Llama-2-70B, GPT-3.5 and longformer (fine-tuned on the training set) on the original HotpotQA dev set (ori) and our adversarially constructed examples (adv). All the tests for the LLMs are done in the few-shot chain of thought prompt setting. EM and F1 Performance Scores are reported. F1 scores are further broken down by (left to right): the number of "fake" paragraphs; whether "fake" paragraphs are related; the type of entity modified, if adversarial paragraphs are unrelated, and if both the adversarial paragraphs are generated from the second sub-question of two different fake sub-question pair.

| Model | Overall | | Paragraph Count | | Paragraph Related | | Modified Type | | | Second Sub-Q only | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | adv | ori | 2 | 4 | Yes | No | Named | Other | No | Yes | No | adv |
| Llama-2-13B | 23.6 | 50.4 | 34.8 | 15.6 | 54.1 | 40.4 | 13.6 | 33.6 | 12.0 | 68.1 | 48.7 | 19.7 |
| Mixtral-8x7B | 40.4 | 67.7 | 53.2 | 18.5 | 52.7 | 24.4 | 82.1 | 36.5 | 11.1 | 67.9 | 50.1 | 17.7 |
| Llama-2-70B | 46.2 | 65.3 | 49.7 | 21.1 | 51.3 | 23.6 | 80.2 | 62.6 | 17.6 | 36.6 | 7.5 | 69.0 |
| GPT-3.5 | 52.0 | 69.8 | 59.7 | 16.1 | 80.1 | 51.9 | 25.2 | 84.6 | 22.2 | 6.5 | 60.4 | 51.3 |
| longformer | 40.5 | 72.9 | 49.7 | 22.1 | 35.8 | 16.6 | 69.5 | 49.5 | 26.6 | 70.5 | 55.4 | 15.1 |

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

Table 6 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), Zellers et al.(2019)].

## 5.3 Analysing the effects of different parameters

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

### 5.3.1 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 6, "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.

#### 5.3.2 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 6, 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.

#### 5.3.3 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 6, 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.

#### 5.3.4 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 an-

swering 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 6, "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".

## 6 Conclusion

We explored whether LLMs can perform multi-hop reasoning when presented with seemingly plausible yet ultimately incorrect reasoning paths. To do so, we conducted an extensive evaluation to show how LLMs' multi-hop reasoning abilities differ from the previous generation of PLM-based NLP methods relying on fine-tuning. We found that existing adversarial attacks are inadequate to probe the capabilities of LLMs; thus we introduced a simple yet powerful framework based on generating paragraphs that contain seemingly plausible yet wrong alternative reasoning chains, compatible with any benchmark that requires multi-hop reasoning. Our extensive empirical study shows that all evaluated LLMs (including GPT-4) struggle to succeed on the proposed benchmark. The framework facilitates the generation of adversarial paragraphs, enabling the creation of more rigorous tests which could lead to more robust models. Datasets augmented with such adversarial paragraphs could allow the models to move away from learning non-robust features like basic lexical matching and enable improved reasoning capabilities. We release data and code to the wider research community on Github: `https://github.com/zawedcvg/Are-Large-Language-Models-Attentive-Readers`.

## Limitations

The main limitation of the proposed method is that it requires the question to be broken down into its sub-questions. Specifically, we use Tang2021's SubQA dataset, but existing question decomposition techniques like Min2019b and Perez2020 can be used to adapt the framework to all HotpotQA questions or any other dataset that deals with multi-hop reasoning. Furthermore, we use the same algorithm for all types of questions to generate seemingly plausible alternate reasoning paths. However, datasets such as HotpotQA distinguish between different types of multi-hop reasoning, e.g. bridge and comparison. Relying on this knowledge, more sophisticated methods to create seemingly plausible alternate reasoning paths could be developed. Although we perform extensive tests to ensure that the current method generates adversarial paragraphs that do not contradict the gold paragraphs, there is no formal guarantee for it.

## References

[AlKhamissi et al.(2022)] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A Review on Language Models as Knowledge Bases. arXiv:2204.06031.

[Bowman(2022)] Samuel R. Bowman. 2022. The Dangers of Underclaiming: Reasons for Caution When Reporting How NLP Systems Fail. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1484–1499.

[Brown et al.(2020)] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

[Chiang and Lee(2024)] Cheng-Han Chiang and Hung-Yi Lee. 2024. Over-Reasoning and Redundant Calculation of Large Language Models.

[Chiang et al.(2024)] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua

Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv:2403.04132.

[Chomsky(1965)] Noam Chomsky. 1965. *Aspects of the theory of syntax*, volume 11. MIT Press, Cambridge, MA.

[Cobbe et al.(2021)] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

[De Marneffe et al.(2014)] Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

[Deng et al.(2024)] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Benchmark Probing: Investigating Data Leakage in Large Language Models. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning*.

[Devlin et al.(2019)] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

[Ding et al.(2021)] Jiayu Ding, Siyuan Wang, Qin Chen, and Zhongyang Wei. 2021. Reasoning Chain Based Adversarial Attack for Multi-Hop Question Answering. arXiv:2112.09658.

[Dua et al.(2019)] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2368–2378.

[Gardner et al.(2020)] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco,

Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1301–1323.

[Gururangan et al.(2018)] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 107–112.

[He et al.(2023)] Zexue He, Yu Wang, An Yan, Yao Liu, Eric Y. Chang, Amilcare Gentili, Julian McAuley, and Chun Nan Hsu. 2023. MedEval: A Multi-Level, Multi-Task, and Multi-Domain Medical Benchmark for Language Model Evaluation. In *EMNLP 2023*, pages 8725–8744.

[Huang et al.(2023a)] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. Large Language Models Can Self-Improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068.

[Huang et al.(2023b)] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhou, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023b. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv:2311.05232.

[Inoue et al.(2020)] Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. R4C: A Benchmark for Evaluating RC Systems to Get the Right Answer for the Right Reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750.

[Jiang and Bansal(2019)] Yichen Jiang and Mohit Bansal. 2019. Avoiding Reasoning Shortcuts: Adversarial Evaluation, Training, and Model Development for Multi-Hop QA. arXiv:1906.07132.

[Kintsch(1988)] Walter Kintsch. 1988. The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, 95(2):163–182.

[Lewis et al.(2020)] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

[Li et al.(2024)] Hao Li, Yuping Wu, Viktor Schlegel, Riza Batista-Navarro, Tharindu Madusanka, Iqra Zahid, Jiayan Zeng, Xiaochi Wang, Xinran He, Yizhi Li, and Goran Nenadic. 2024. Which Side Are You On? A Multi-Task Dataset for End-to-End Argument Summarisation and Evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*.

[Liu et al.(2019)] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

[Liu et al.(2023)] Boyang Liu, Viktor Schlegel, Riza Batista-Navarro, and Sophia Ananiadou. 2023. Argument Mining as a Multi-Hop Generative Machine Reading Comprehension Task. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10846–10858.

[Madusanka et al.(2023)] Tharindu Madusanka, Iqra Zahid, Hao Li, Ian Pratt-Hartmann, and Riza Batista-Navarro. 2023. Not All Quantifiers Are Equal: Probing Transformer-Based Language Models' Understanding of Generalised Quantifiers. In *EMNLP 2023*, pages 8680–8692.

[Min et al.(2019a)] Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. Compositional Questions Do Not Necessitate Multi-Hop Reasoning. arXiv:1906.02900.

[Min et al.(2019b)] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Multi-Hop Reading Comprehension Through Question Decomposition and Rescoring. In *ACL*.

[OpenAI(2023)] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

[Perez et al.(2020)] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised Question Decomposition for Question Answering. arXiv:2002.09758.

[Perez et al.(2023)] Ethan Perez, Sam Ringer, Kamile Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu,

Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kerner, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, Jared Kaplan, and A. I. Surge. 2023. Discovering Language Model Behaviors with Model-Written Evaluations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 13387–13434.

[Qi et al.(2020)] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

[Reimers and Gurevych(2019)] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP 2019*, pages 3982–3992.

[Sakarvadia et al.(2023)] Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. 2023. Memory Injections: Connecting Multi-Hop Reasoning Failures During Inference in Transformer-Based Language Models. In *BlackboxNLP 2023*, pages 342–356.

[Saparov et al.(2023)] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the General Deductive Reasoning Capacity of Large Language Models Using OOD Examples. In *NeurIPS '23*, pages 3083–3105.

[Schlegel et al.(2020)] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. 2020. A Framework for Evaluation of Machine Reading Comprehension Gold Standards. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5359–5369.

[Schlegel et al.(2021)] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. 2021. Semantics Altering Modifications for Evaluating Comprehension in Machine Reading. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13762–13770.

[Schlegel et al.(2022a)] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. 2022a. A Survey of Methods for Revealing and Overcoming Weaknesses of Data-Driven Natural Language Understanding. *Natural Language Engineering*, pages 1–31.

[Schlegel et al.(2022b)] Viktor Schlegel, Kamen V. Pavlov, and Ian Pratt-Hartmann. 2022b. Can Transformers Reason in Fragments of Natural Language? In *EMNLP 2022*, pages 11184–11199.

[Shi et al.(2023)] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schürli, and Denny Zhou. 2023. Large Language Models Can Be Easily Distracted by Irrelevant Context. arXiv:2302.00093.

[Sun et al.(2023)] Hao Sun, Alihan Hüyük, and Mihaela van der Schaar. 2023. Query-Dependent Prompt Evaluation and Optimization with Offline Inverse RL. In *The Twelfth International Conference on Learning Representations*.

[Tang et al.(2021)] Yuxuan Tang, Hwee Tou Ng, and Anthony K. H. Tung. 2021. Do Multi-Hop Question Answering Systems Know How to Answer the Single-Hop Sub-Questions? In *EACL*.

[Touvron et al.(2023)] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

[Trivedi et al.(2020)] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is MultiHop QA in DiRe Condition? Measuring and Reducing Disconnected Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8846–8863.

[Vaswani et al.(2017)] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

[Wang et al.(2023)] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171.

[Wei et al.(2023)] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

[Welbl et al.(2018)] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing Datasets for Multi-Hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

[Yang et al.(2024)] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do Large Language Models Latently Perform Multi-Hop Reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 10210–10229.

[Yang et al.(2018a)] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

[Yang et al.(2018b)] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning.

2018b. HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. arXiv:1809.09600.

[Zellers et al.(2018)] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104.

[Zellers et al.(2019)] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

# A System Prompt for Q/A task

You are a helpful, respectful, and honest question-answering assistant. You will be given a context and a question. Answer the question using only the context. You will break the questions into sub-questions. You will then use these sub-questions to get to the final answer. The final answer must have "Final Answer:" prepended to it. Thus your output will be in the following format:

Sub-question 1: [subquestion 1] Answer: [answer 1]
Sub-question 2: [subquestion 2] Answer: [answer 2]
...
Sub-question n: [subquestion n] Answer: [answer n]
Final Answer: [final answer]

The final answer should be limited to 5 words with just the answer and no explanation/information.

# B System Prompt for Creating Fake Paragraphs

You are a helpful, respectful and honest assistant. You will be given two questions. For each of the questions, you will generate a fake named entity of the appropriate type. The fake entity should not be the same as any of the two questions and the fake answer. The answer and information should not be related to any real-life entity. The paragraphs generated must match the tone of the given two paragraphs. Furthermore, the two paragraphs generated must not contradict any of the information in the supporting paragraphs provided by the user.

Use the fake answer generated for the first question to replace all instances of "[answer]" in the second question. Use the newly generated question and generate a fake answer for it. Ensure that the fake answer generated is not the same as any

of the provided words you need to avoid. Similar to the first question, use the fake answer and the question to generate a fake paragraph. You will generate the fake paragraphs as if they were part of a Wikipedia article. You must maintain a neutral and informative tone.

Generate the two paragraphs as separate articles about 75–100 words each. All the answers and paragraphs must be made up of fake names and fake information. The information/names should not reference anyone in real life. Generate exactly one paragraph for each question. Remember to replace all instances of "[answer]" with the answer from the first question and adjust the paragraphs accordingly. However, you must not mention the fact that the details/entities in the paragraphs are fake/imaginary.

## C Dependency Type Definitions

Table 7 consists of definitions of the dependency relations used in the attack. All the definitions are based on DeMarneffe2014.

## D Reproducibility

The parameters used for fine-tuning the longformer model are:

- Batch size: 64
- Learning Rate: $3 \times 10^{-5}$
- Train Epochs: 3
- Maximal answer length: 30

All experiments were carried out on a single NVIDIA Titan RTX GPU.

## E User Study to Verify Adversarial Paragraphs

To verify that examples don't influence gold labels, a user study was conducted involving 5 participants, all of whom had at least college level education. Each participant received the same random sample of 49 questions from the adversarial dataset. For each question, participants were provided with relevant lines from the gold paragraph and relevant lines from adversarial paragraphs intended to distract from the correct answer.

Each question had 4–5 options. One was the correct answer, two were the answers to the sub-questions for the plausible paragraphs, and the rest were titles

Table 7: Definitions based on Universal Dependencies

| Term | Definition |
|---|---|
| nsubj | nominal subject: a nominal which is the syntactic subject and the proto-agent of a clause. |
| nsubj:pass | passive nominal subject: a noun phrase which is the syntactic subject of a passive clause. |
| obl | oblique: a nominal (noun, pronoun, noun phrase) functioning as a non-core (oblique) argument or adjunct. |
| obj | direct object: the noun phrase which is the (accusative) object of the verb. |
| acl:relcl | adnominal relative clause: a clause modifying some head (typically a noun) that is understood to fulfill some grammatical role in the RC. |
| appos | appositional modifier: a nominal immediately following the first noun that serves to define, modify, name, or describe that noun. |
| amod | adjectival modifier: any adjective or adjectival phrase that serves to modify the meaning of the nominal. |
| nmod | nominal modifier: nominal dependents of another noun or noun phrase functioning as an attribute or genitive complement. |
| compound | noun compounds. |
| flat | flat structure: used to combine elements of an expression where none of the immediate components can be identified as the sole head. |

of the "supporting facts" from HotpotQA if these were not already included in the options. After each question, the user was asked if the two sources of information contained contradicting information.

Table 8 shows three different metrics:

- Average Accuracy: The percentage of questions answered correctly by the participants.
- Accuracy-Combined: A question is given 1 point if more than 3 participants answered it correctly, and 0.5 points if exactly 2 participants answered correctly and no incorrect answer got more than 2 votes.
- Accuracy-UB: Adapted from the HotpotQA paper, this metric checks if any

of the users were able to answer a particular question correctly.

Table 8: The three different metrics for accuracy

| Metric | Value |
|---|---|
| Average Accuracy | 70.6% |
| Accuracy-Combined | 84.6% |
| Accuracy-UB | 95.9% |

Table 9 shows the number of questions marked as contradictory with a confidence level of greater than 40%. Only 5 questions were marked as contradictory with confidence $> 40\%$, and upon inspection just one of them was found to have contradicting information. We attribute the marking of these questions as contradictory to human error and waning attention. Through the accuracy metrics and the count of questions marked as contradictory under different confidence levels, we can conclude with certainty that these distractors do not affect the gold labels and are not an issue for humans.

Table 9: The confidence level of a question being contradictory

| Confidence Level | Count of Contradictory |
|---|---|
| 40% | 5 |
| 60% or more | 0 |

# F Do Existing Techniques Make Models More Robust?

Table 10 shows the results of running more advanced prompting methods than naive chain-of-thought reasoning, such as instructed chain-of-thought prompting [Shi et al.(2023)] and self-consistency [Wang et al.(2023)] on the Llama-2-13B and Llama-2-70B. The setting is 2 plausible paragraphs with the modifiable portion as "other". While self-consistency leads to a smaller decrease in F1 score under the attack, the gains in robustness (4.2 F1 points) are limited. Instruct prompting on the other hand doesn't provide any relevant improvements. This suggests that our findings unveil a behaviour of LLMs that cannot be corrected simply by using more advanced prompting techniques.

Table 10: Effect of self-consistency on F1 score

| Model | Original | adv |
|---|---|---|
| Llama-2-13B CoT | 49.6 | 23.9 |
| Llama-2-13B CoT+Instruct | 40.7 | 21.5 |
| Llama-2-70B CoT | 60.0 | 49.6 |
| Llama-2-70B CoT+Instruct | 59.6 | 49.8 |

## G Performance of SOTA LLM

To see how well our method generalises to better models, we evaluate GPT-4, the best-performing LLM at the time of writing. GPT-4 was tested on 250 examples (due to cost constraints), where fake paragraphs are related by alternate reasoning chains, using Named Entity as the main entity type and alternating between two and four fake paragraphs. Table 11 shows that GPT-4 is more resilient to the attack as compared to the other LLMs that were tested. However, it still exhibits a drop of 14 points in F1 under the strongest adversarial attack setting i.e., related paragraphs, four adversarial paragraphs.

Table 11: F1 scores of GPT-4 for 2 and 4 fake paragraphs

| Setting | Original | adv |
|---|---|---|
| 2 fake paragraphs | 87.1 | 79.9 |
| 4 fake paragraphs | 87.1 | 77.0 |