Deceptive Semantic Shortcuts on Reasoning Chains: How Far Can Models Go without Hallucination?

Bangzheng Li[1] **Ben Zhou**[2] **Fei Wang**[3] **Xingyu Fu**[2] **Dan Roth**[2] **Muhao Chen**[1]
[1]University of California, Davis [2]University of Pennsylvania
[3]University of Southern California
bzhli@ucdavis.edu

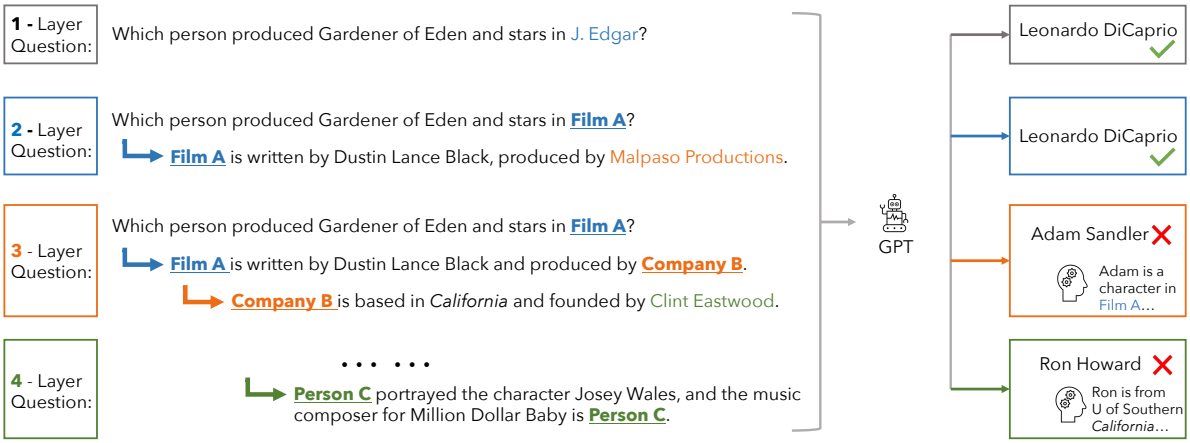

Figure 1: LLMs make errors when correct surface-level semantic cues – entities – are recursively replaced with descriptions, and the errors are likely related to *token similarity*. GPT-3.5-turbo is used for this example.

Abstract

Despite the high performances of large language models (LLMs) across numerous benchmarks, recent research has unveiled their suffering from hallucinations and unfaithful reasoning. This work studies a type of hallucination induced by semantic associations. We investigate to what extent LLMs take shortcuts from certain keyword/entity biases in the prompt instead of following correct reasoning paths. To quantify this phenomenon, we propose a novel probing method and benchmark called EureQA. EureQA is an entity-searching task where a model finds a missing entity based on described multi-hop relations with other entities. These deliberately designed multi-hop relations create deceptive semantic associations, and models must stick to the correct reasoning path instead of incorrect shortcuts to find the correct answer. Experiments show that existing LLMs cannot follow correct reasoning paths and resist the attempt of greedy shortcuts, with GPT-4 only achieving 62% accuracy. Analyses provide further evidence that LLMs rely on semantic biases to solve the task instead of proper reasoning, questioning the validity and generalizability of current LLMs' high performances.[1]

[1]Code and data are available at https://github.com/VincentLeebang/eureqa

1 Introduction

Recent progress of large language models (LLMs) has showcased their emergent abilities in a wide range of reasoning tasks, encompassing competence in program synthesis (Kuznia et al., 2022), mathematical reasoning (Mishra et al., 2022), symbolic logic reasoning (Gaur and Saunshi, 2023), and common-sense reasoning (Feng et al., 2023). Innovations in inference methods have further improved language models' reasoning capabilities, either by generating intermediate steps toward the ultimate solution (Wei et al., 2022) or by decomposing complex inquiries into manageable subproblems (Zhou et al., 2022a; Yao et al., 2023). Research has further demonstrated that when incorporated with these techniques, LLMs have achieved exceptional results on multi-hop and complex QA benchmarks (Geva et al., 2021).

Nonetheless, it remains under-investigated whether the effectiveness of LLMs' reasoning pattern (OpenAI, 2023; Team, 2023) is genuinely based on a sensible reasoning path or if these models predominantly depend on semantic associations (i.e., word distributions from the pre-training data) to generate a plausible answer. Fig. 1 shows how

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7675–7688
June 16-21, 2024

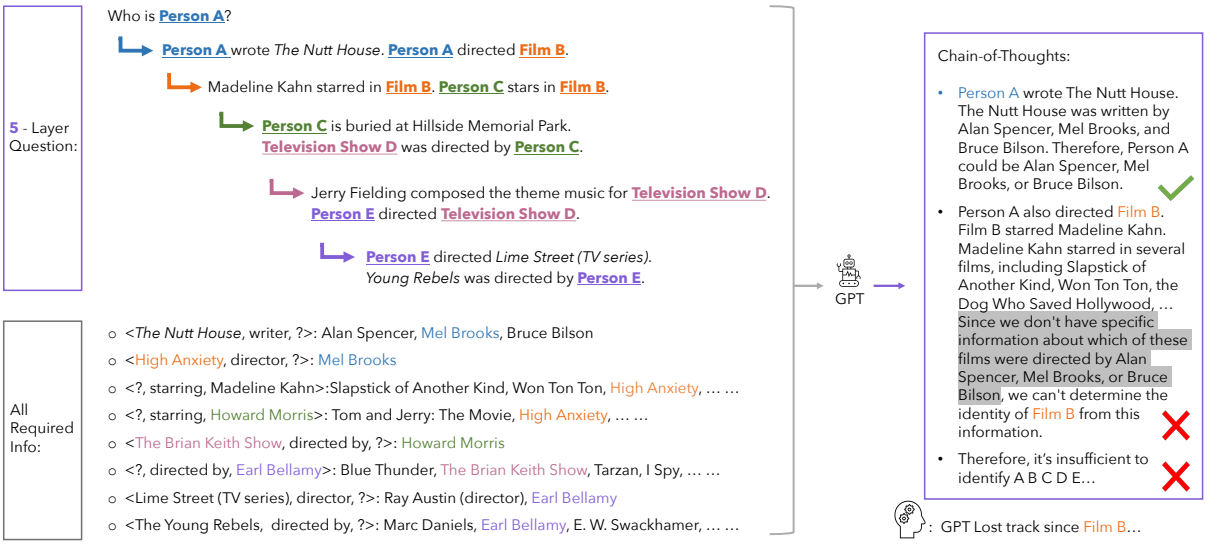

Figure 2: Even given all the required information needed for the question (selected information shown in the figure), GPT-4 still makes mistakes starting early layers (highlighted in grey). We only show partial output here. Notice that we give GPT few-shot prompts.

ChatGPT[2] fails to find the missing entity as we gradually remove relevant semantic cues as the reasoning depth grows. The original question (labeled as "1 Layer" in Fig. 1) asks models to find an individual that satisfies two distinct conditions. It can be resolved by finding the set of people who are producers of the film "Gardener of Eden" and the set of cast members of "J. Edgar", and checking for the overlapping entity. However, language models such as ChatGPT may not necessarily perform this sensible entity-seeking process. This is because the correct answer can simply (but incorrectly) be inferred by the high occurrence of "Leonardo DiCaprio" in the model's training data with "Gardener of Eden" and "J. Edgar".[3] In such a scenario, ChatGPT may generate the correct response through this shortcut of semantic association. Using semantic shortcuts such as co-occurring entities works well on common test cases. Still, they limit systems' generalizability and robustness on cases not aligning with pre-training distributions. This is evident if we remove entities directly relevant to the answer, such as "J. Edgar", and replace them with recursive entity-seeking problems where models must first identify them, shown as "3 Layers" in Fig. 1. We observe that ChatGPT no longer produces the right answer without enough semantic hints and hallucinates "Adam Sandler", which may be because "Eden" is related "Adam", and at the same time, both the writer and main character of "Gardener of Eden" is named "Adam." In addition, ChatGPT almost always comes up with "Adam Sandler" if asked for actors named "Adam."

To investigate and quantify whether LLMs can follow a correct reasoning path and resist the attempt to take "greedy shortcuts" during inference, we introduce EUREQA (Extending Underlying reasoning Chains in QA), a sophisticated multi-hop question answering dataset, meticulously designed for diminishing semantic associations and gauging the capability of LLMs in undertaking extensively chained reasoning processes. To create EUREQA, we design a method that removes semantic shortcuts that will lead to the correct answer while adding "deceptive semantic cues" that are distracting and irrelevant to the correct answer. We also guarantee a simple and consistent reasoning path and only use common facts that should be memorized by the models. In this way, if an LLM fails on such tasks, it directly suggests that the LLM is taking incorrect shortcuts during inference. For instance, we modify the original question in Fig. 1 by substituting "J. Edgar" with a placeholder dubbed "Film A" and supplementing a descriptive sentence about the film. Our intent with this strategy is to lessen the model's reliance on direct information from the film name. We can automatically extend the depth of a reasoning chain by further replacing entities in the descriptive sentence about "Film A"

[2]In this paper, we consistently utilize the gpt-3.5-turbo-0301 checkpoint for ChatGPT.

[3]In fact, if we prompt ChatGPT with only three keywords "Gardner of Eden," "J. Edgar" and "actor" and let it generate anything, the outcome is almost always "Leonardo DiCaprio."

with corresponding abstract names. In particular, this generation method is developed on a knowledge base, ensuring that the information employed predates the training data of the examined LLMs.

We then evaluate model performances on the EUREQA benchmark. We show that state-of-the-art LLMs,[4] are incapable of proper reasoning for instances in EUREQA, with GPT-4 only achieving 62% to 64% accuracy in identifying common entities in Wikipedia and ChatGPT less than 40%. On the contrary, humans achieve near-perfect performances without much effort because of the simplicity and consistency of the gold reasoning path. We also show that GPT-4 performance strongly correlates with the semantic similarities between the gold answer and other entities mentioned in the question. These findings combined suggest that even the best language models, with detailed in-context-learning (ICL) processes, still fall for deceptive semantic shortcuts and hence hallucinate and fail on EUREQA.

To summarize, the contribution of our work is three-fold. First, we propose a novel method for generating question-answering data with extended reasoning chains, which allows us to create deceptive semantic shortcuts. Second, we propose EUREQA, a QA dataset specifically designed for evaluating LLMs in scenarios with reduced or deceptive semantic associations. Finally, our extensive experimental findings suggest that contemporary LLMs predominantly depend on semantic associations in question answering. As entities in question are replaced with alternate inquiries, LLMs cannot adhere to an accurate reasoning process necessary for problem resolution.

2 EUREQA

In this section, we introduce the dataset EUREQA for evaluating LLMs on extended reasoning chains. We start with the definition and generation approach of a reasoning chain (§2.1), followed by the processes for question generation (§2.2) and refinement (§2.3).

2.1 Reasoning Chain

In EUREQA, every question is constructed through an implicit reasoning chain, as depicted in Fig. 3 (1). This chain is structured in layers, each of which comprises three components: an **entity** e_i, an associated **fact** f_i about e_i, and a relation r_i which connects e_i and the entity in the succeeding layer of the chain, referred to as e_{i+1}. The identity of e_i is constrained by f_i, e_{i+1} and r_i together. In most cases, neither f_i nor e_{i+1}/r_i can lead to e_i, but we guarantee the uniqueness of e_i when subjecting to both constraints. This design allows us to extend the reasoning chain by replacing e_i with f_i, r_i, and e_{i+1}, until we provide the actual entity name of e_N. A valid reasoning path would be to take the provided e_N and resolve e_{N-1} based on f_{N-1} and r_{N-1} until we identify e_1, which is the answer to the overall question.

To automatically collect such data, we use a structured knowledge base DBpedia (Auer et al., 2007), because it provides accurate facts and relations between entities, and at the same time, all these facts and relations should have been memorized by most LLMs since Wikipedia is seen during pre-training. We use the *2021-09* DBPedia snapshot to guarantee that the information of entities was accessible to LLMs during their training phase. The criteria for choosing valid e_i, r_i, and f_i to form chains are the following:[5]

Entity. Each entity e_i must exist in the knowledge base as an "entity."

Relation. Each relation r_i should connect e_i with more than one potential candidate entities e_{i+1}. This ensures no direct semantic shortcut between e_i and e_{i+1}. For example, the relation "`award`" is deemed valid for the entity "Tiger Woods" because he has won multiple awards. In contrast, the relation "`college`" is invalid because DBPedia only has one corresponding entity, which is *Stanford University*.

Fact. Facts are generated based on DBpedia relations and destination entities for each e_i. Specifically, a valid fact f_i consists of a relation r_i^f and a destination entity e_i^f. Note that r_i^f is different from r_i, because it is only used to generate a single factual statement regarding e_i, and not being used to extend the reasoning chain. We adopt two criteria for fact selection, namely `easy` and `hard`. The `easy` criterion selects (r_i^f, e_i^f) pairs where e_i is the only entity in the database that corresponds to the triplet (*?entity*, r_i^f, e_i^f). For example, *Jason Connery* is the only entity that fits into (*?entity*, `father`, *Sean Connery*) because *Jason Connery* is the only child of *Sean Connery*. In this way, f_i itself is

[4] In this context, ChatGPT and GPT-4.

[5] Being "valid" here refers to satisfying the following criteria for data extraction.

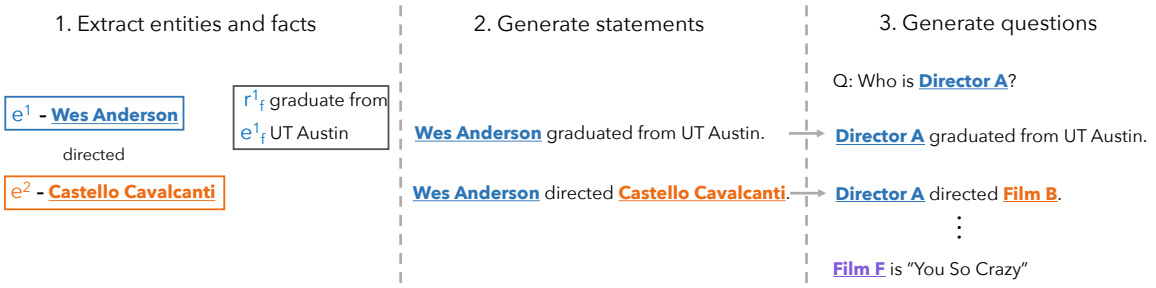

Figure 3: The data generation process of EUREQA.

sufficient for identifying e_i. Contrarily, the `hard` selection ensures that not only e_i satisfies (*?entity*, r_i^f, e_i^f). For instance, (`parentCompany`, *Comcast*) is a `hard` fact for "*NBCUniversal*" since "*Comcast*" is also the parent company of "*Xfinity*." The `hard` criterion ensures that f_i cannot uniquely identify e_i. We apply `easy` and `hard` criteria to fact generation respectively for EUREQA, resulting in two distinct sets of data: EUREQA $^{\text{easy}}$ and EUREQA $^{\text{hard}}$.

Chain Construction. Following the previously established criteria, we employ a random-walk algorithm to fabricate chains of reasoning. The procedure initiates with a designated seed entity e_1 in the knowledge base. Subsequently, a relation r_1, an associated fact f_1, and a subsequent entity e_2 are chosen randomly to construct a valid layer of the reasoning chain. This operation is continuously executed until no more valid layer is found or the cumulative number of layers equals N_{max}. We set $N_{max} = 5$ in this work. We perform at most 50 random walks starting from each seed entity to optimize efficiency and remove duplicated chains. Seed entities are sourced via two methods, either through prompting ChatGPT or by data extraction from https://today.yougov.com/.

This approach ensures that reasoning chain generation for EUREQA satisfies the following three properties of each chain:

Reasoning-dependent. Every intermediate layer relies on information from the subsequent layer in the chain for resolution. Consequently, the model must commence from the terminal layer and engage in sequential reasoning throughout the chain.

Length-flexible. The extent of the reasoning chain can be conveniently adjusted by adding or removing layers, thereby enabling an evaluation of the depth of reasoning.

Determinism-adjustable. Determinism of the reasoning chain can be altered by omitting fact f_i in a layer, facilitating an evaluation of the model in addressing questions with multiple potential answers, which is not considered in this work.

2.2 Question Generation

In EUREQA, each question, denoted as q, is a natural language articulation of the above reasoning chains. A layer-by-layer procedure translates this structured chain into a human-readable text. Here we denote $q = q_0 + q_1 + ... + q_n + m_{n+1}$ where $\{q_i | 1 <= i <= n\}$ indicates the sub-question for the i-th layer and "+" stands for concatenation. q_0 and m_{n+1} will be introduced later in this subsection. Fig. 3 (2) provides a tangible example of this translation process. Every layer comprises an entity e_i, a relation r_i, a succeeding entity e_{i+1}, and an associated fact f_i, which includes the corresponding r_i^f and e_i^f. The initial step involves translating the triplets, (e_i, r_i e_{i+1}) and (e_i, r_i^f, e_i^f) into two statements, $s_i^{relation}$ and s_i^{fact}, respectively. This is done by few-shot prompting ChatGPT, which we detail in §8.

The subsequent step involves substituting each entity e in every statement with a placeholder, as illustrated in Fig. 3 (3). In particular, each layer now associates with a pair of statements, namely $s_i^{relation}$ and s_i^{fact}, which derive from the translation of the relational triplet (e_i, r_i e_{i+1}) and the factual triplet (e_i, r_i^f, e_i^f) correspondingly. In this context, we proceed to solely obfuscate e_i and e_{i+1} in $s_i^{relation}$ and s_i^{fact} by substituting them with their respective hypernyms, h_i and h_{i+1}, whilst preserving the identity of e_i^f. Hypernym details are extracted from DBpedia if available and sought from the LLM otherwise. ChatGPT is employed in our approach with two few-shot examples, listed in §8, obtained from DBpedia, to ensure a consistent level of granularity of hypernym. Each hypernym is subsequently appended with a specific label that is assigned in alphabetical order. For example, the third layer of a reasoning chain could hold "Actor C stars in TV Series D" as $s_3^{relation}$ while s_3^{fact} could

be "Actor C is the artist for *Along on Christmas Day*." Here, e_3 is substituted by "Actor C" and e_4, which also appears in the fourth layer, is replaced by "TV Series D." The resulting statements with masked entities are denoted as $m_i^{relation}$ and m_i^{fact} respectively.

The final step involves generating an interrogative question, represented as q_0, and an entity information statement, denoted as m_{n+1}, pertaining to q_n. The selection of the initial interrogative question is contingent upon the hypernym of the entity e_1 in the first layer. Specifically, q_0 starts with "Who is ..." if the hypernym h_1 corresponds to a person or "What is ..." in all other cases. In the example shown in Fig. 3, "Wes Anderson" has the hypernym "director." The generation process for m_{n+1} is essentially uncomplicated. Given that $m_i^{relation}$ in the terminal layer exhibits the relation between e_n and e_{n+1}, and no additional layers exist to identify e_{n+1}, we generate m_{n+1} utilizing the template "h_{n+1} is e_{n+1}.", which is exemplified as "Film F is *You So Crazy*." in Fig. 3.

2.3 Question Refinement

We perform **viability filtering** to check if a reasoning chain can be correctly followed to derive the expected answer. As mentioned above, each layer of a specific chain specifies e_i by both a relational triplet (e_i, r_i e_{i+1}) and a factual triplet (e_i, r_i^f, e_i^f). By excluding e_i from these triplets, we query our knowledge base with both ($?variable$, r_i e_{i+1}) and ($?variable$, r_i^f, e_i^f). The layer is deemed viable if e_i is a unique solution of $?variable$. We only retain those reasoning chains where each layer has passed the viability filtering.

This work aims to analyze models' reasoning capabilities. As such, we employ a **knowledge filtering** procedure to ensure that most LLMs have sufficient world knowledge to answer our questions. To illustrate this, we consider a question, $q = q_0 + q_1 + ... + q_n + m_{n+1}$, and verify both $s_i^{relation}$ and $s_i^{relation}$, where $1 <= i <= n$. In more explicable terms, we evaluate knowledge by presenting ChatGPT with the prompt "*Is this statement correct: [s_i] Yes or No?*" A statement is considered as memorized if the LLM response includes a "Yes."[6] A q_i is deemed to have satisfied the examination criteria if and only if *both* of $s_i^{relation}$ and $s_i^{relation}$ are categorized as memorized.

[6]we applied self-consistency strategy (Wang et al., 2022b) with three runs for a majority vote.

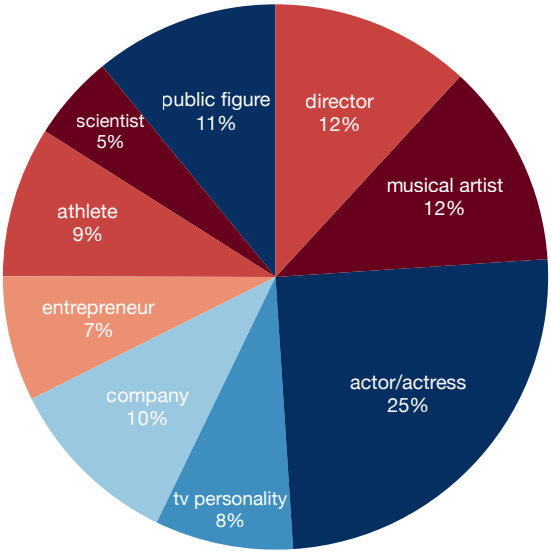

Figure 4: Categorical distribution of seed entities in questions of EUREQA.

Difficulty	easy	hard	hard	hard	hard	hard
#Layers	5	5	4	3	2	1
Count	428	1363	300	300	300	300

Table 1: Statistics of EUREQA.

2.4 Data Statistics

As indicated in Tab. 1, EUREQA encompasses a total of 2,991 questions. The dataset is split into two levels of difficulty: `hard` and `easy` according to the criteria discussed in §2.2. Specifically, the `easy` set comprises 428 five-layer questions while the `hard` set includes a larger set of 1,363 five-layer questions. This `easy` version enables manifesting LLM behavior towards questions with sufficient semantic shortcuts. From this `hard` set, we randomly select 300 five-layered questions. By removing layers sequentially, the `hard` set also yields 300 questions in each number of layers from four to one respectively. These `hard` questions with varying layers allow for a comprehensive assessment of LLMs in terms of their reasoning capabilities across various depths of reasoning. The distribution of categories of seed entities for the questions in EUREQA is shown in Fig. 4, showcasing a broad spectrum of topics encompassed by these entities, effectively reducing any potential bias arising from specific entity categories.

depth	hard										easy	
	d=1		d=2		d=3		d=4		d=5		d=5	
	direct	*icl*	*direct*	*icl*	*direct*	*icl*	*direct*	*icl*	*direct*	*icl*	*direct*	*icl*
ChatGPT	22.3	53.3	7.0	40.0	5.0	39.2	3.7	39.3	7.2	39.0	13.1	47.0
Gemini-Pro	45.0	49.3	29.5	23.5	27.3	28.6	25.7	24.3	17.2	21.5	30.6	38.9
GPT-4	60.3	76.0	50.0	63.7	51.3	61.7	52.7	63.7	46.9	61.9	66.4	81.8

Table 2: Accuracy of ChatGPT, Gemini-Pro and GPT-4 across different depths d of reasoning (number of layers in the questions) as well as the difficulty of the questions. We evaluate two prompt strategies: *direct* zero-shot prompt and *icl* with two examples.

3 Experiment Setup

This section presents the experiment configurations employed for assessing the long-chain reasoning capabilities of LLMs through EUREQA. This section begins with a discussion on model configurations (§3.1), subsequently introducing the adopted prompting methodologies (§3.2), and finally, it scrutinizes the evaluation configurations (§3.3).

3.1 Model Configuration

We consider state-of-the-art LLMs to evaluate on EUREQA: ChatGPT (gpt-3.5-turbo-0301), Gemini-Pro (Gemini 1.0 Pro) and GPT-4 (gpt-4-0314). All models run with a temperature τ of 0.8.

3.2 Prompting Methods

We employ two prompting methods in our experiment: namely, *direct* and *icl*. The *direct* technique presents raw questions to the model without any supplementary context. On the other hand, *icl* method utilizes a few-shot approach, preceding each query with two context-specific examples. Each example is characterized by a sample problem, succeeded by the statement "Let's solve this question step by step" and a step-by-step solution written by humans. In these solutions, the problem is analyzed from the final layer through the initial layer to give the correct answer, thereby obviating any necessity for backtracking during the problem-solving procedure. An example is detailed in §8.

3.3 Evaluation Protocol

We evaluate the **accuracy** of LLMs towards the correct answer. Considering that EUREQA functions as a free-form QA benchmark and our observation indicates that LLMs typically respond with comprehensive reasoning processes, following prior studies (Agrawal et al., 2015; Ossowski and Hu, 2023), we employ a string-match criterion: If the correct answer, identifiable as an entity, is present in the LLM response, we deem such a response as correct. We use a self-consistency of five runs with ChatGPT and Gemini-Pro, but we do not apply self-consistency to GPT-4 due to cost issues.

4 Results

The results of the state-of-the-art LLMs on EUREQA, as reported in Tab. 2, illustrate the variations in performance across different depths of reasoning and levels of difficulty. In general, with the entities recursively substituted by the descriptions of reasoning chaining layers, and therefore eliminating surface-level semantic cues, these models generate more incorrect answers. When the reasoning depth increases from one to five on hard questions, there is a notable decline in performance for all models, with an average accuracy decrease of 14.2% when using the *icl* prompt and 14.3% when using the *direct* prompt. The performance of LLMs is significantly higher on the easy set compared to the hard set. This is evidenced by a marked increase in accuracy, averaging 7.0% for ChatGPT, 15.4% for Gemini-Pro, and 19.7% for GPT-4. This finding underscores the significant impact that semantic shortcuts have on the accuracy of responses, and it also indicates that GPT-4 is considerably more capable of identifying and taking advantage of these shortcuts.

It can also be inferred from the results that the demonstrated human-written examples do help improve the performance. For both ChatGPT and GPT-4, using the *icl* prompt consistently leads to better performance than the *direct* prompt. This suggests that these models can benefit from examples that provide explicit and well-crafted human-written reasoning processes. These few-show examples encourage the models to resolve the questions with the correct reason instead of guessing with semantic cues. Nonetheless, even with clear reasoning processes provided, models still tend to

fail for incorrect shortcuts.

Another observation is that LLMs do not always perform worse on questions with more depth of reasoning. For instance, ChatGPT with *direct* prompt is 3.5% more accurate when the depth of reasoning d increases from four to five. We believe this also indicates that LLMs do not always follow the reasoning path and may instead answer the question based on surface-level semantic shortcuts. Further analyses can be found at §5.2

To identify the errors in these LLM responses, we randomly sampled 20 data points where GPT-4 with *icl*, the best performing method in this benchmark, has generated incorrect answers. The most frequent error pattern lies in hallucination during the reasoning process, encompassing 80% of the cases. In such cases, an intermediate reasoning stage erroneously conceived an inaccurate, albeit plausible answer, culminating in accumulated errors in the final answer. For instance, the "founder of a rocket company" was misconstrued as "*Elon Musk*, the founder of SpaceX", rather than the correct answer of "*Jeff Bezos*, the founder of Blue Origin." This phenomenon proves our claim that LLM relies on semantic cues for problem-solving instead of cognitive reasoning.

5 Analysis and Discussions

In this section, we perform further analyses and provide a more in-depth discussion of our findings.

5.1 Can human solve EUREQA?

Towards a more comprehensive insight into human proficiency on EUREQA, a human analysis was carried out focusing on the `hard` set characterized by a reasoning depth of five. This evaluation involves two computer science PhD students as the annotators to ensure their expertise. A selection of 50 questions, randomly extracted from the set, was subjected to annotation. Each question received annotations from both annotators independently, ensuring a dual perspective on every query. During the annotation process, annotators are presented with an input question to which they are required to provide an answer. To facilitate this, they are granted access to conduct information searches through the *dbpedia-snapshot-2021-09* database. This could be achieved either through visiting the specific entity page or by querying via a SPARQL portal. The averaged accuracy of annotators achieves 95% with an Inter-annotator Agreement of Cohen $\kappa = 0.79$.

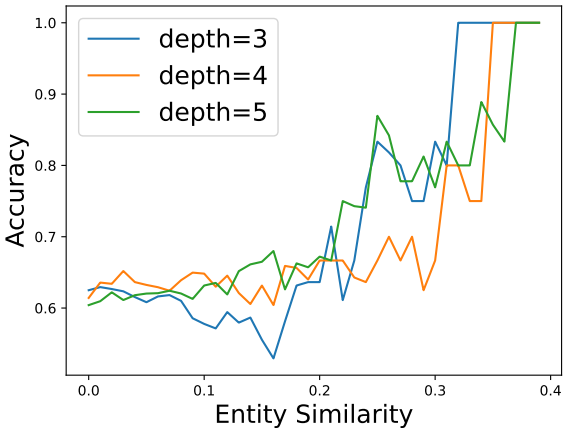

Figure 5: The correlation between GPT-4 performance on EUREQA hard set and entity similarities.

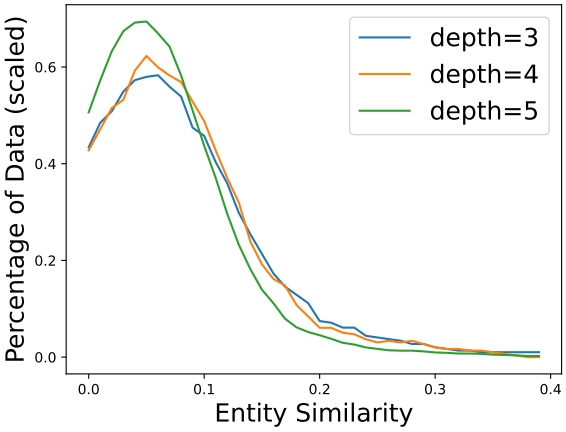

Figure 6: The distribution of entity similarity scores.

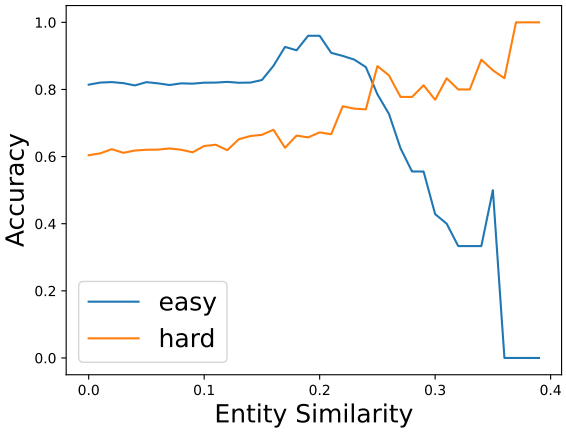

Figure 7: GPT-4 performances with different entity similarity scores between the easy and hard sets.

5.2 Do LLMs take Shortcuts?

One of our main motivations is to test if language models can follow a simple yet effective reasoning

chain instead of taking semantic shortcuts based on entity associations. We design an analytical experiment based on entity similarity to investigate whether LLMs take such semantic shortcuts. Our intuition is to model the correlation between the performances and the averaged semantic similarities between the gold answer and other entities mentioned in the question. In an ideal situation where a model takes an optimal reasoning path for each instance, we will see a uniform distribution of performances: the accuracy for various degrees of entity similarities will be relatively the same. If the model relies much on entity biases and takes semantic shortcuts, we will see an increasing performance when the mentioned entities are more closely related to the target answer.

To infer entity similarities, we employ an off-the-shelf sentence Transformer model[7] and encode the entity strings into embeddings. With such embeddings, we calculate the dot-product similarity between the target answer and all other Wikipedia entities mentioned in the question and compute an averaged similarity for each instance. We then draw the correlation curve of model performances on instances with certain similarities.[8] This method derives a relatively continuous curve based on our limited evaluation data.

Observation 1. Fig. 5 shows that GPT-4's performance positively correlates with the entity similarities in the instances. This shows that the model relies on semantic shortcuts instead of following the correct reasoning path.

Observation 2. GPT-4 performs similarly on different depths if the entity similarity scores are the same, especially on instances with lower similarities (< 0.25). This again suggests that the model relies on spurious entity biases to solve the question, because it does not do better on instances with shorter reasoning paths. That being said, the performance differences between lower and higher depths mostly come from the different data distributions, as shown in Fig. 6.

Observation 3. Fig. 7 shows another interesting finding, where GPT-4 does better on easy instances with lower similarity scores (< 0.25) and performs worse on those with higher similarity scores. Our explanation is that GPT-4 tends to take more obvious shortcuts by relying on only one or two entities in the easy instance, and "early exits" without even considering the entities that are closely relevant to the gold answer.

5.3 Do open source LLMs perform better?

To expand the scope of our findings, we experimented with open-sourced LLAMA-2 models across different sizes on hard questions of EUREQA and their results can be found at Tab. 3. Similar to our observations on GPT-series models, there's a notable decline in the accuracy of Llama models as the reasoning depth increases from one to five on the hard set. We've also conducted the entity similarity analysis which led to the same observation as Observation 1 in §5.2: LLMs' performance positively correlates with the entity similarities in the instances. These conclusions strengthened our claim that models rely on semantic shortcuts instead of following the correct reasoning path.

5.4 Will prompting solve EUREQA?

Although the effectiveness of prompting techniques is out of the scope of this paper, we still additionally tested the Tree of Thought(TOT)(Yao et al., 2023) method on ChatGPT with a "propose" strategy, which tries to decompose the questions layer by layer and solve them sequentially. Our results show that such a prompting method fails completely even if we provide human-written examples for question decomposition. In no experiment did the TOT method generate a valid answer in the final response.

That being said, it is unlikely that there exists a prompting technique that will significantly improve models' performance on EUREQA over CoT, as long as we do not enforce the correct reasoning chain (which defeats the purpose of testing generalizability) during inference.

5.5 Will optimal retrieval solve EUREQA?

Although our knowledge filtering process has already removed the knowledge barrier, it can still be pointed out that whether Retrieval Augmented Generation(RAG) method can solve this task. To address such concerns, we tested GPT-4 on 300 randomly sampled 5-layer hard questions. To minimize the impact of the retrieval method, we investigate a "performance upper bound" setting, which

[7] https://huggingface.co/sentence-transformers/msmarco-distilbert-cos-v5.

[8] Specifically, we start with a similarity value $x_{start} = 0.0$ and increment it by a step of 0.01. For each x_{start}, we find instances that have an averaged similarity score between $[x_{start}, x_{start} + 0.1]$, and compute the accuracy on these instances.

directly injects the retrieval result of the visible entities and relations in the input question from DBpedia. To be specific, statements like "`Storm Warning (1951 film) stars Actor D`" will have "`<Storm Warning (1951 film), starring, ?>: Ginger Rogers, Steve Cochran, Ronald Reagan, Doris Day`" prepended to the input question. The retrieved knowledge will obtain at most 20 candidate entities and the correct entity is guaranteed to be included. A complete example of a question and retrieved knowledge can be found in Fig. 2. We believe this represents an upper bound of the retrieval result. Additionally, we provide each input with a 1-shot demonstration with the retrieved knowledge, input questions, and human-written reasoning thoughts.

The accuracy of GPT-4 through the above process is 62.0%, which is close to our reported performance of 61.9% in the original setting. We can then hypothesize that simply injecting knowledge into the model can hardly solve the problem and the bottleneck remains at the reasoning/question decomposition ability of LLMs. Moreover, it can be concluded that our knowledge-filtering process can effectively estimate parametric knowledge, and it is also reliable after self-consistency.

6 Related Work

We discuss two topics of works that are highly relevant to this study.

Hallucination. Existing research on analyzing LLMs hallucination mainly focuses on three aspects: input-conflicting hallucination, context-conflicting hallucination, and fact-conflicting hallucination (Zhang et al., 2023). Input-conflicting hallucination happens when the LLMs outputs are divergent from prompts (Maynez et al., 2020). Self-contradictory outputs can occur in long-form or multi-turn answers, implying that LLMs lose track of the core input information during generation (Liu et al., 2023). Factual-based errors – misaligns with established world knowledge – are most widely studied since they can happen in various LLMs to affect multiple tasks' performance (Min et al., 2023; Li et al., 2023). Knowledge conflicts also lead to hallucination when in-context information contradicts what LLMs memorize from pre-raining (Longpre et al., 2021; Zhou et al., 2023; Wang et al., 2023). However, none of these existing works look into hallucination on reasoning-required questions and test whether the models are following the correct reasoning path versus the semantic token bias.

Reasoning capability of LLMs. The success of in-context learning (Brown et al., 2020) and instruction tuning (Wei et al., 2021) have inspired various works to solve reasoning tasks by prompting LLMs. More advanced prompting strategies have been proposed to enhance the reasoning capabilities of LLMs. These investigations take advantage of the heuristic nature of human problem-solving procedures and incorporate them into textual prompts as guidance for the models. An example of this can be seen in chain-of-thought prompting (Wei et al., 2022), a method in which intermediate steps are generated leading up to a final solution. Yao et al. (2023) and Zhou et al. (2022b) proposed to decompose the questions into simpler, manageable sub-problems as a method to facilitate complex reasoning. Researchers also attempted to investigate LLMs' reasoning ability over structured knowledge (Ding et al., 2023) which necessitates reasoning with uncertainty as well as back-tracking. Studies prior to LLMs also propose to generate rationales to improve with more faithful reasoning (Rajani et al., 2019; Wang et al., 2022a). Nonetheless, this work questions the exhibited "reasoning capabilities" of LLMs by distinguishing them from the attempt to take greedy semantic shortcuts during the reasoning process.

7 Conclusion

This paper proposes a novel QA benchmark for probing LLM hallucinations induced by semantic shortcuts on reasoning chains. We introduced a systematic method for generating question-answering data with extended reasoning chains. This dataset enables us to examine the reasoning capabilities of LLMs on extended reasoning paths and our experimental results indicate that LLMs predominantly depend on semantic shortcuts for reasoning and such behavior contributes to its failure. Our analyses questioned the validity of current LLMs and also inspired future studies. The problem-solving or reasoning trajectories of humans could be incorporated into the training or inference phase of current Language Modeling systems for more effective computational models.

Ethical Considerations

Innovations in technology often encounter the moral challenge of dual-use: the same develop-

ment can bring both benefits and risks. With the probing method and benchmark presented in this paper, the line between beneficial and harmful usage largely depends on data. Proper utilization of the technology necessitates the legal and ethical acquisition of input text corpora and other modalities of inputs. Legal frameworks and standards are crucial for ensuring proper data use and for granting individuals the right to remove their data. In the absence of such regulation, the ethical use of data depends on the responsibility of technology users. Additionally, the generated and analysis data may exhibit biases that systematically affect accuracy for less represented groups or in new areas, potentially resulting in performance disparities among sub-populations based on ethnicity, race, gender, and other factors. Moreover, the effectiveness of trained systems diminishes when applied to new data that deviates from their training set. Therefore, issues of generalizability and fairness must be thoroughly examined when implementing the methodologies discussed in this paper. It is crucial to embed ethical considerations as fundamental principles at each stage of system development, ensure high levels of transparency and clarity in data, algorithms, models, and functions within the system, release software under open-source licenses to facilitate public scrutiny and investigate strategies to safeguard at-risk groups.

Limitations

Our work proposes an analytical framework and dataset to evaluate how well language models can take the right reasoning paths instead of deceptive semantic shortcuts. To this end, we identify the following limitations.

Limited Baselines. We only consider two variants of language models as our baselines. With more efforts in the future, we can extend to more families of large language models with different prompting techniques. Although we project that all current systems will be far behind human performances on EUREQA, we may identify specific models and methods that are more resistant to deceptive semantic shortcuts.

Limited Entities. We only consider popular entities in Wikipedia as our target answer. Future works may benefit from considering a wider range of entities and a more general setting of our data extraction processes.

Acknowledgement

We appreciate the reviewers for their insightful comments and suggestions. Bangzheng Li is supported by the Faculty Startup Fund of UC Davis, and the Provost's Fellowship. Fei Wang is supported by the Annenberg Fellowship and the Amazon ML Fellowship. Ben Zhou was funded by ONR Contract N00014-23-1-2365. Xingyu Fu was funded by NSF grant IIS-2212433. Muhao Chen is supported by the NSF Grant IIS 2105329, the NSF Grant ITE 2333736, the Faculty Startup Fund of UC Davis, a Cisco Research Award and two Amazon Research Awards.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4 – 31.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wenxuan Ding, Shangbin Feng, Yuhan Liu, Zhaoxuan Tan, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. Knowledge crosswords: Geometric reasoning over structured knowledge with large language models.

Yu Feng, Ben Zhou, Haoyu Wang, Helen Jin, and Dan Roth. 2023. Generic temporal reasoning with differential analysis and explanation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12013–12029, Toronto, Canada. Association for Computational Linguistics.

Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5889–5903, Toronto, Canada. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle

use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Less is more: Summary of long instructions is better for program synthesis. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4532–4552, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv e-prints*, pages arXiv–2305.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5807–5832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Timothy Ossowski and Junjie Hu. 2023. Retrieving multimodal prompts for generative visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2518–2535, Toronto, Canada. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, Singapore. Association for Computational Linguistics.

PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022a. Pinto: Faithful language reasoning using prompt-generated rationales. In *The Eleventh International Conference on Learning Representations*.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate problem solving with large language models.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Ben Zhou, Kyle Richardson, Xiaodong Yu, and Dan Roth. 2022a. Learning to decompose: Hypothetical question decomposition based on comparable texts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2223–2235, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022b. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

8 Appendices

We present the prompts used in data generation or model evaluation on EUREQA. During the question generation process (§2.2), The Verbalization Prompt in Fig. 8 transforms a triplet, $(e_i, r_i\ e_{i+1})$ or (e_i, r_i^f, e_i^f), into a statement by querying ChatGPT with it. Typing Prompt in Fig. 9 is later used when we try to substitute an entity with its hypernyms in a statement but its hypernym is unavailable from DBpedia. Fig. 10 is one of the two examples we used in the *icl* prompt answering questions with a depth of five. Questions with fewer depths of reasoning adopt two similar examples with corresponding depths. Fig. 11 is an example input of the RAG experiment, as elaborated in §5.5 and illustrated in Fig. 2.

Tab. 3 is the experimental results of the open-sourced Llama-2 models on EUREQA, as discussed in §5.3.

Translate the triplets into a sentence. The second entity is the relationship between the first and third entities in the triplets.

triplet: [Robert_De_Niro, foundedBy, TriBeCa_Productions]

sentence: TriBeCa_Productions is founded by Robert_De_Niro.

triple: [Frederic_L._Smith, almaMater, University_of_Michigan]

sentence: Frederic_L._Smith's alma mater is University_of_Michigan.

triple: [{ent1}, {rel}, {ent2}]

sentence:

Figure 8: Verbalization Prompt: verbalize (*entity*, `relation`, *entity*) into a natural sentence.

Define entities with their hypernyms:

Steven_Spielberg is a director.

General_Motors is a car company.

{entity} is a

Figure 9: Typing Prompt: generate hypernym of an entity when it is unavailable from DBpedia.

Question: Who is Person A? Person A produced Gardener of Eden. Person A starred in Film B. Brian Grazer is the producer of Film B. Company C is the production company of Film B. Invictus (film) was produced by Company C. Person D founded Company C. Person D starred in Every Which Way but Loose. Person D produced Film E. Malpaso Productions produced Film E. Person F stars in Film E. Person F is Brian Dennehy.

Let's solve this question step by step:

Brian Dennehy stars in Film E, which is produced by Malpaso Productions. Among all films starring Brian Dennehy, only The Stars Fell on Henrietta is produced by Malpaso Productions. So Film E is The Stars Fell on Henrietta. Therefore Person D starred in Every Which Way but Loose and produced The Stars Fell on Henrietta. The producers of The Stars Fell on Henrietta include Clint Eastwood and David Valdes. Only Clint Eastwood starred in Every Which Way but Loose, so Person D is Clint Eastwood. Invictus (film) was produced by Company C, which was founded by Clint Eastwood, confirming that Company C is Malpaso Productions. Brian Grazer is the producer of Film B and Malpaso Productions is the production company of film B. Given that the only satifying film is J._Edgar, Film B is identified as J._Edgar. Person A produced Gardener of Eden and starred in J._Edgar. J._Edgar is produced by Brad Simpson and Leonardo DiCaprio. Only Leonardo DiCaprio starred in J._Edgar. So Person A is Leonardo DiCaprio. The answer is Leonardo DiCaprio.

Figure 10: One of the two examples used in *icl* prompt, with depth of five reasoning process.

Knowledge:
<?, music, Craig Safan>: Delivering Milo, Remo Williams: The Adventure Begins, Tag: The Assassination Game, ...
<?, starring, Xander Berkeley>: Deadly Dreams, The Last Full Measure (2004 film), Tag: The Assassination Game, ...
<?, spouse, Linda Hamilton>: James Cameron, Bruce Abbott
<Tag: The Assassination Game, starring, ?>: Perry Lang, Linda Hamilton, John Mengatti, Bruce Abbott,...
<Curvature (film), starring, ?>: Linda Hamilton, Lyndsy Fonseca
<Bruce Abbott, spouse, ?>: Linda Hamilton, Kathleen Quinlan
<?, productionCompany, StudioCanal>: 40 Days and 40 Nights, Boys on the Side, Terminator 2: Judgment Day, ...
<?, starring, Linda Hamilton>: Holy Water (film), Terminator 2: Judgment Day, Smile (2005 film), A Girl Thing, Shadow Conspiracy, ...
<The Art of Avatar, author, ?>: James Cameron, Jon Landau, Peter Jackson
<Terminator 2: Judgment Day, director, ?>: James Cameron

Question: Who is Person A? Person A is the author of The Art of Avatar. Person A directed Film B. StudioCanal produced Film B. Person C starred in Film B. Person C is starring in Curvature (film). Person D is Person C's spouse. Linda Hamilton is the spouse of Person D. Person D stars in Film E. Craig Safan composed the music for the film Film E. Person F stars in Film E. Person F is Xander Berkeley.

Figure 11: Sample knowledge and question in the "performance uppper bound" setting of RAG study

depth	d=1		d=2		d=3		d=4		d=5	
	direct	*icl*	*direct*	*icl*	*direct*	*icl*	*direct*	*icl*	*direct*	*icl*
Llama-2-7b	29.5	22.0	21.4	11.9	15.8	13.0	10.6	11.3	10.9	14.5
Llama-2-13b	40.0	28.5	27.0	16.4	20.5	15.8	11.3	15.9	13.9	19.4
Llama-2-70b	45.0	57.0	24.5	35.2	17.8	39.0	18.5	32.5	12.1	34.5

Table 3: Accuracy of Llama-2 models across different depths d of reasoning (number of layers in the questions). We evaluate two prompt strategies: *direct* zero-shot prompt and *icl* with two examples. The 7b and 13b versions are chat variants and the 70b version is instruct variant.