Semantic Inversion, Identical Replies: Revisiting Negation Blindness in Large Language Models

Jinsung Kim*, **Seonmin Koo***, **Heuiseok Lim**†
Department of Computer Science and Engineering, Korea University
{jin62304,fhdahd,limhseok}@korea.ac.kr

Abstract

Large language models (LLMs) often fail to capture semantic changes in queries due to negation, and generate incorrect responses. Negation frequently exists in the real world and is useful for understanding the opposite or absence of a statement, so it is an essential element in logical reasoning. Previous studies have explored LLMs' ability to capture negations 'separately' from their ability to properly ground knowledge for positive queries. However, this perspective is limited in that it cannot clearly distinguish whether the cause of incorrect responses is the logical incoherence caused by negations or the lack of grounding ability for the given context. To address this issue, we focus on the phenomenon of the model failing to capture semantic contradictions in negated queries despite its accurate understanding of knowledge about positive queries. We term this phenomenon *negation blindness* on the query. We propose a verification framework that includes task design and measurement methods to verify this issue. In detail, we establish two criteria for systematic task design–i) 'complexity' and ii) 'constrainedness'–and devise four verification tasks accordingly. Moreover, we analyze the results extensively and provide insights into problem alleviation feasibility through experiments on various approaches[1].

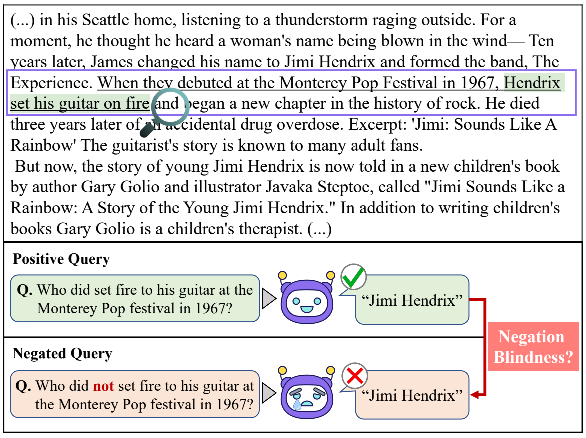

Figure 1: Actual generation example of Mistral-7B regarding the *negation blindness* problem in given positive and negated queries.

1 Introduction

One of the major challenges is that large language models (LLMs) can still generate inaccurate responses for given contexts and user queries (Ji et al., 2023; Chen et al., 2023; Rozner et al., 2024). LLMs often easily generate incorrect information for negation scenarios that cause semantic changes in the text, such as 'not' (Varshney et al., 2024; Asher and Bhar, 2024). Negation is a common occurrence in the real world and is also essential for logical reasoning, as it helps understand the opposite or absence of a statement (MacDonald, 1965; Barker and Jago, 2012; Arnaout and Razniewski, 2023). Focusing only on situations based on ideal positive queries leads to a gap between research and these real-world needs, so extensive studies on negation are needed.

To this end, recent studies have explored the generation ability of LLMs for negative queries, focusing on showing that the model's answering accuracy for negative queries deteriorates (Hosseini et al., 2021; Jang et al., 2022; Arnaout et al., 2022; Truong et al., 2023; Asher and Bhar, 2024).

However, while prior studies have evaluated the model's handling of negation, they typically assess this ability in isolation from its capacity to resolve corresponding positive-form queries. This decoupled perspective limits interpretability, as it becomes difficult to determine whether inaccurate responses arise from inadequate grounding of contextually relevant knowledge or from logical failures triggered by negation. For instance, when a model produces an incorrect response to a negated

* Equally contributed. † Corresponding author.
[1]Our code and resources can be found at https://www.github.com/jin62304/NegationBlindness.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21434–21471
November 4-9, 2025 ©2025 Association for Computational Linguistics

query, yet also fails on the associated positive query, the error likely stems from a general lack of grounding ability. In such cases, the effect of negation itself cannot be precisely attributed. Therefore, a more fine-grained analysis necessitates disentangling errors caused by insufficient grounding from those due to logical incoherence by negation.

To this end, we adopt a pairwise verification that jointly examines the model's grounding ability on a positive query and its sensitivity to the corresponding negated form. We focus on a phenomenon we term *negation blindness*, wherein a model fails to reverse its inference when presented with a negated query, despite having correctly resolved its affirmative counterpart. As illustrated in Figure 1, this behavior is prevalent in LLM outputs: although models often demonstrate accurate knowledge grounding for positive queries, they frequently overlook the semantic inversion by negation, resulting in logically incoherent responses.

Therefore, this study is conducted with the following research question: **How can it be verified that LLMs suffer from *negation blindness* despite understanding the given context?**

Accordingly, we propose a verification framework, including task designs and measurement methods, to verify LLMs' *negation blindness* problem. In detail, we establish two criteria–i) complexity and ii) constrainedness–and devise the following four tasks accordingly: boolean selection, multiple-choice selection, cloze-style completion, and free-form generation. Through extensive experiments and analysis, we investigate the *negation blindness* problem under various factors such as different models, parameter sizes, and composition of exemplars. Moreover, we provide insights by experimenting with various approaches, such as multi-agent debate, to explore the feasibility of mitigating the observed problem.

The *negation blindness* problem reveals a deeper issue of logical coherence beyond task accuracy, showing that models often mishandle semantic inversion despite knowing the positive counterpart. This underscores coherence as a critical evaluation dimension and positions BLD score as a useful diagnostic for selecting models in tasks where logical consistency is essential.

Our contributions are threefold: (1) We conduct a comprehensive study of the *negation blindness* problem, where logical coherence is degraded by negation, even though LLMs can adequately ground knowledge of its positive counterpart. (2) We propose a verification framework for the problem, including task design and measurement methods, and probe various LLMs based on this. (3) We provide extensive experimental results and analysis on the verification, as well as insights into the potential of various methods to alleviate the problem.

2 Related Work

The concept of negation has been considered an important factor in verifying the logical reasoning ability of language models (MacDonald, 1965; Barker and Jago, 2012; Arnaout and Razniewski, 2023). This is because the model's lack of awareness of negation can lead to misunderstanding sentence intent and generating inaccurate responses (Minsky, 1997). Therefore, investigating cases where the failure to recognize negation becomes severe is an essential foundation for model improvement (Morante et al., 2011).

Even before the advent of LLM, attempts have been conducted to assess the model's understanding of negation. Hosseini et al. (2021), Gubelmann and Handschuh (2022), and Jang et al. (2022) focus on the effects of negation in tasks such as masked knowledge retrieval and natural language inference. Some studies address the effect of model size on the comprehension of negation (Truong et al., 2023). In particular, Kassner and Schütze (2020) primarily examined negation within cloze-style probes (e.g., "Bird can [MASK]."), which are narrow in scope and limited in downstream applicability. Also, they restricted their evaluation to encoder-only, small to medium-scale PLMs (e.g., BERT, ELMo, Transformer-XL). In contrast, our study introduces the expanded design allowing systematic investigation of negation handling across tasks of varying nature and difficulty, thereby providing more comprehensive insights into LLM behavior, and also extends verification to a broader and more contemporary range of models.

Despite the substantial parameter sizes in the LLM era, several studies have presented that LLMs still generate inappropriate responses to negative knowledge and negation. Varshney et al. (2024) and Asher and Bhar (2024) analyze the impact of negation on the LLM's generation accuracy in various tasks. Following studies that various prompting methods are effective in improving reasoning ability, there have been attempts to apply them to negation situations (Jang et al., 2023; Varshney et al., 2024). In addition, there are studies that reveal the

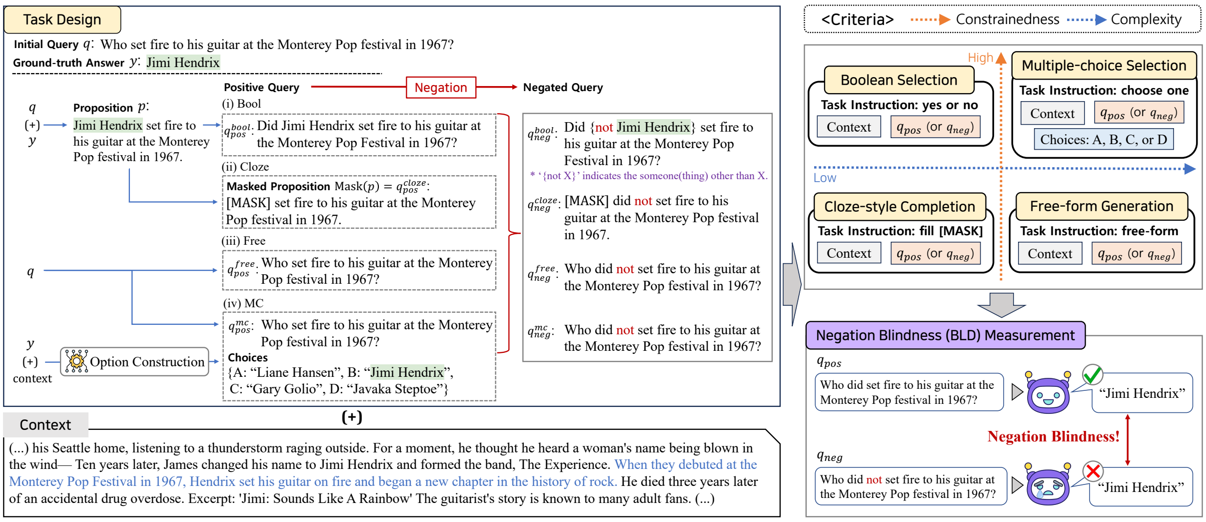

Figure 2: Overview of the proposed framework to verify *negation blindness* problem.

phenomenon that models have difficulty describing knowledge they already know negatively, and that when negative knowledge is input, the generation ability may be reduced due to the bias of LLMs (Arnaout et al., 2022; Chen et al., 2023).

However, while prior work on negation in LLMs has primarily focused on isolated negated queries, it often overlooks the logical relationship between positive and negated counterparts. As a result, such approaches fall short in thoroughly evaluating a model's logical coherence with respect to negation. Specifically, the absence of pairwise analysis impedes the ability to determine whether prediction failures on negated queries stem from insufficient knowledge utilization or from logical inconsistency. To overcome these limitations, this study proposes a fine-grained evaluation framework that jointly assesses the model's grounding in positive knowledge and its sensitivity to negation.

3 Verification Framework

This section describes the proposed framework, which includes the design of verification tasks and their evaluation schema. First, we describe the basic denotations for the tasks (§ 3.1), and then we make a statement about the *negation blindness* problem we raised (§ 3.2). Afterward, we establish criteria for systematically designing detailed verification tasks (§ 3.3), and accordingly present detailed task design and corresponding measurement method based on the criteria (§ 3.4). Figure 2 demonstrates an overview of the proposed verification framework for the *negation blindness* problem.

3.1 Denotations

This section describes the basic denotations for the QA task and problem statement. We denote the context required to answer a question as ctx, the initial question, which is one of the elements from the set of all initial questions Q, as q, and the ground-truth answer to q as y.

The initial query q is modified to q_s^t according to each verification task type t, where $t \in \{bool, mc, cloze, free\}$[2], and s indicates whether the sentence type is positive or negated ($s \in \{pos, neg\}$). The negated query q_{neg} is built by conducting negation to the positive-form query q_{pos}. For example, if q_{pos} is "Who was the President of the United States in 2020?", then q_{neg} is "Who wasn't the President of the United States in 2020?". The model M aims to take as input a pair (ctx, q_s^t) and generate an appropriate response y_s^t corresponding to the task type t and the polarity s (i.e., positive or negative).

An important aspect of y_s^t lies in the semantic inversion introduced by negation. Specifically, incorporating negation into the text semantically reverses the intent of the query, which in turn leads to an inversion of the language model's prior probability distribution over possible answers (Asher and Bhar, 2024). Accordingly, the distribution of the negated answer y_{neg} can be interpreted as the complement of the distribution of the positive answer y_{pos}: $\mu(y_{neg}|\text{input } x) = \{1 - \mu(y_{pos}|\text{input } x)\}$. That is, under the principle of logical coherence for negation, given the ground-truth answer y_{pos} to

[2] A detailed description of each task is given in Section 3.4.

the positive query, then the correct answer y_{neg} for the negated query q_{neg} must lie within the distribution representing "not y_{pos}."

Further theoretical and empirical clarifications regarding the semantic validity of our proposed extension to negated scenarios are provided in Appendix A[3].

3.2 Problem Statement: *Negation Blindness*

This study defines the *negation blindness* problem as LLM's behavior that (1) adequately addresses a positive query, but (2) fails to capture semantic inversion by the addition of negations, resulting in the same responses in both situations.

Equation 1 defines the set V_s as the collection of queries q_s^t for which the model's response $M(ctx, q_s^t)$ is semantically equivalent to the ground-truth answer y_{pos}^t corresponding to a positive query:

$$V_s = \left\{ q_s^t \,\middle|\, M(ctx, q_s^t) = y_{pos}^t \right\}, \tag{1}$$

where V_{neg} means the set of queries where the model fails to capture the semantic inversion induced by negation—specifically, cases in which the model attempts to generate y_{pos}^t in response to a negated query q_{neg}, thereby ignoring the intended logical contrast between q_{neg} and the corresponding positive query q_{pos}.

Accordingly, BLD score, which quantifies *negation blindness*, can be formalized as follows:

$$BLD = \frac{|V_{pos} \cap V_{neg}|}{|Q|}. \tag{2}$$

3.3 Criteria Setup

This study establishes the following two criteria for designing tasks that systematically verify the *negation blindness* problem: i) complexity and ii) constrainedness. Regarding complexity, we follow Bloom's taxonomy (Krathwohl, 2002), a well-known taxonomy in the cognitive domain, to subdivide 'tasks with relatively more considerations' and 'tasks with fewer considerations'. Also, we divide verification tasks into 'selection' and 'generation' tasks according to the constrainedness.

For example, tasks that require choosing an answer only from given options are more constrained than tasks that generate responses in a free format. Also, binary classification, which has only two options, may have lower complexity than multiple-choice, which has four or more options.

[3] Specifically, we further demonstrate whether the logical framework of distributional inversion under negation, as proposed in prior work by Asher and Bhar (2024), can be effectively extended to the negation scenarios addressed in our study.

3.4 Task Design

Based on the two established criteria, this study classifies the verification tasks into the following four: boolean selection, multiple-choice selection, cloze-style completion, and free-form generation.

Boolean Selection. In the boolean selection (hereinafter Bool) task, the model should select an appropriate response between `true` or `false` for a given q^{bool}. According to the criteria outlined in Section 3.3, this task is characterized by (high constrainedness, low complexity).

The procedure for constructing the Bool task is as follows: First, a proposition p in the form of a declarative sentence is built based on the initial QA pair (q, y). For example, when q is "What is the capital of China?" and y is "Beijing", p becomes "The capital of China is Beijing." Then, to generate a positive query q_{pos}^{bool}, p is converted into a question form without an interrogative pronoun. Depending on the syntactic structure, the 'to be' verb or auxiliary verb is placed at the beginning of the sentence.

Moreover, to compose a negated query q_{neg}^{bool}, we implement semantic inversion by prepending the negation component to the position corresponding to y in q_{pos}^{bool}. For example, when q_{pos}^{bool} is "Is Beijing the capital of China?", q_{neg}^{bool} becomes "Is {not Beijing} the capital of China?". If the model answers "Yes." to both q_{pos}^{bool} and q_{neg}^{bool}, it is regarded as a *negation blindness* case.

In particular, in the Bool task, it is crucial to negate the entity corresponding to the answer y (e.g., '`{not entity}`'), rather than the predicate itself (e.g., 'isn't'). This design choice mitigates interpretive ambiguity by explicit negation (e.g., 'not'), which can confound the evaluation of model responses. For instance, the questions "Is Beijing the capital of China?" and "Isn't Beijing the capital of China?" may both elicit the response "Yes," depending on pragmatic interpretation, despite differing syntactic structures (Celce-Murcia et al., 1983).

Multiple-choice Selection. In the multiple-choice selection (hereinafter MC) task, the model aims to select an appropriate option from a set of options $\mathcal{O}$ given to q. While this task exhibits a high level of constrainedness similar to the Bool task, it is more complex due to the larger number of candidate options.

q^{mc}_{pos} is set to be the same as the initial query q, and based on this, a negation term is appended to the predicate part of q^{mc}_{pos} to obtain q^{mc}_{neg}. For example, when q is "What is the capital of China?", q^{mc}_{pos} and q^{mc}_{neg} are "What is the capital of China?" and "What is <u>not</u> the capital of China?", respectively.

Algorithm 1 Option Set Construction Algorithm

Input: initial query q, ground-truth answer y, context ctx, and NER model $\mathcal{N}$

Extract set of entities $\mathcal{E}_y$ from y using $\mathcal{N}$
Extract set of entities $\mathcal{E}_{ctx}$ from ctx using $\mathcal{N}$
Remove duplicates and ensure $y \notin \mathcal{E}_{ctx}$
Split $\mathcal{E}_{ctx}$ into $\mathcal{E}_{\text{same}}$ and $\mathcal{E}_{\text{diff}}$ based on entity type match with y
Randomly sample k entities from $\mathcal{E}_{\text{same}}$
if sampled set contains fewer than k entities **then**
  Sample the remaining entities from $\mathcal{E}_{\text{diff}}$
end if
Construct option set $\mathcal{O}$ by combining y with the sampled entities
Shuffle $\mathcal{O}$ and determine i^*, index of y in $\mathcal{O}$

Output: Option set $\mathcal{O}$ and ground-truth index i^*

In addition, constructing an option set $\mathcal{O}$ is required for the MC task, and Algorithm 1 outlines the procedure. We obtain a synthetic option set $\mathcal{O}$ consisting of (k+1) choices through the process. That is, $\mathcal{O}$ consists of a ground-truth answer y and k additional options (distractions) considering the entity type of y. Also, we randomly arrange the position i^* of y in $\mathcal{O}$ to reduce the position bias.

Cloze-style Completion. In the cloze-style completion (hereinafter Cloze) task, the model aims to generate an appropriate phrase at position `[MASK]` within a given masked proposition $MASK(p)$. Compared to the selection tasks described above, this task has no fixed answer candidates and relatively low constraints. Also, since it infers only the masked part, its complexity is lower than that of a free-form generation.

First, to obtain the positive query q^{cloze}_{pos}, we replace the span of y in proposition p, which is composed based on q and y, with `[MASK]` token. That is, q^{cloze}_{pos} is defined as $q^{cloze}_{pos} = MASK(p)$. Next, we acquire q^{cloze}_{neg} by appending a negation term to the predicate of q^{cloze}_{pos}. For example, when p is "Beijing is the capital of China.", $MASK(p)$ is "`[MASK]` is the capital of China.", and q^{cloze}_{neg} is "`[MASK]` is <u>not</u> the capital of China.". If the model generates semantically identical phrases to q^{cloze}_{pos} and q^{cloze}_{neg}, the *negation blindness* has occurred.

Free-form Generation. In the free-from generation (hereinafter Free) task, the model generates a response to a given query q^{open}_s without being restricted by a specific format (low constrainedness). The complexity is relatively high because the number of possible answers is large. The method of composing q^{open}_{pos} and q^{open}_{neg} is the same as the MC task ("What is the capital of China?" and "What is <u>not</u> the capital of China?"). If the model generates responses with the same meaning for q^{open}_{pos} and q^{open}_{neg}, it is determined as *negation blindness* case.

4 Experimental Setup

Detailed experimental setups, including the tools and hyperparameters, can be found in Appendix B, while additional analyses such as qualitative results are provided in Appendix C. For the prompt templates, please refer to Appendix D.

Models. The proposed framework utilized LLM-as-a-Labeler for the data construction, including query negation, employing GPT-4o mini and GPT-4.1 mini[4] models, which are gpt-4o-mini-2024-07-18 and gpt-4.1-mini-2025-04-14 versions, respectively.

Also, for verification experiments on *negation blindness* with the LLMs' responses, we adopted ChatGPT (gpt-3.5-turbo-0125 version) (OpenAI-Blog, 2022), LLaMA3.1-8B (Dubey et al., 2024), Mistral-7B-instruct (Jiang et al., 2023), and Claude3.5-Haiku (Anthropic, 2024).

Dataset and Metrics. TriviaQA (Joshi et al., 2017) dataset was used as the raw source data for the experiment. We construct the data resources for the verification experiments on *negation blindness* by randomly sampling the raw dataset.

To ensure the quality and reliability of the newly curated verification set, we conducted cross-validation using three independent human annotators. In constructing the dataset, we employed a few-shot prompting approach informed by a diverse range of negation scenarios, as illustrated in the prompt templates provided in Appendix D, in order to minimize annotation errors and inconsistencies [5]. For the actual verification experiments,

[4]OpenAI GPT-4o, OpenAI GPT-4.1

[5]Error cases during the construction process can be found in Appendix B.

Metrics	Accuracy			BLD (↓)	Accuracy			BLD (↓)
	pos (↑)	neg (↑)	avg (↑)		pos (↑)	neg (↑)	avg (↑)	
Model	*Selection*							
	Boolean (Bool)				Multiple Choice (MC)			
ChatGPT	75.84	57.02	66.43	**30.90**	94.66	63.20	78.93	35.96
LLaMA3.1	84.55	56.74	70.65	38.48	95.22	31.18	63.20	67.70
Mistral	**91.01**	**66.01**	**78.51**	31.74	95.79	32.87	64.33	67.13
Claude3.5	90.17	47.47	68.82	49.72	**96.91**	**64.61**	**80.76**	**33.99**
Model	*Generation*							
	Cloze-style (Cloze)				Free-form (Free)			
ChatGPT	86.24	49.44	67.84	49.16	**89.33**	55.34	72.33	42.98
LLaMA3.1	86.80	49.44	68.12	49.16	87.92	64.89	76.40	34.27
Mistral	84.27	**55.90**	**70.08**	**42.70**	85.67	**72.19**	**78.93**	27.81
Claude3.5	**89.61**	45.79	67.70	53.37	84.83	69.66	77.25	**26.12**

Table 1: Verification results for the responses of each LLM. 'pos,' 'neg,' and 'avg' indicate the accuracy for positive queries, the accuracy for negative queries, and their average, respectively. 'BLD' means the measurement result of *negation blindness* described in Section 3.2. **Bold** text represents the best performance.

each model was evaluated on a total of 3,256 examples, thereby facilitating consistent and comparable assessment across settings.

To evaluate the phenomenon of *negation blindness*, we introduce the BLD score, as formally defined in Section 3.2. In particular, to compute the BLD score, we employ a textual entailment-based evaluation mechanism to assess the semantic equivalence (i.e., accuracy) between the model's response $M(ctx, q_s^t)$ and the ground-truth answer y_{pos}^t to the positive query q_{pos}^t.

5 Experimental Results and Analysis

In this section, we present the various verification results for the *negation blindness* problem and also examine the feasibility of alleviating the problem.

5.1 Verification Results

Please note that further results and analyses that could not be included in the main text due to space limitations are in Appendix C. For instance, these include results concerning the expansion of benchmark datasets (Appendix C.1), experiments on the diversification of negation cues (Appendix C.2), as well as qualitative analyses (Appendix C.8).

Insight 1: Lower accuracy on negated questions does not necessarily imply more *negation blindness*. Table 1 shows the performance of each model for four tasks. Accuracy refers to the individual performance of models for positive and negated queries, which have been the focus of previous negation-related studies. In contrast, the BLD score measures *negation blindness* problem, where a lower score indicates a model with lower occurrences of *negation blindness* problem (§ 3.2).

First, the Mistral model demonstrates the highest average accuracy in the Bool task, indicating superior individual performance in solving each positive and negated query. However, the BLD score of ChatGPT, which has 12.08 points lower accuracy than that of the Mistral model, stands at 30.9 points, reflecting the least *negation blindness*. This suggests that ChatGPT has less logical incoherence due to negation in the Bool task compared to other models.

Furthermore, in both MC and Free tasks, Claude 3.5 exhibited the lowest degree of *negation blindness*. In the Free task, Mistral outperformed Claude3.5 by at least 1.68 points in accuracy for both positive and negative scenarios. However, notably, the BLD score was 1.69 points lower for Claude 3.5 than for Mistral.

These results suggest that only evaluating the model's accuracy on negation scenarios does not provide sufficient evidence for whether reasoning ability for positive queries leads to the capturing ability of negation. Therefore, bridging the assessment of the *negation blindness* problem with the model's ability to solve positive queries is significant in the field of LLM probing on negation.

Insight 2: Increasing parameter size may lead to a reduction of *negation blindness*. Previous studies have shown that increasing the model's parameter size promotes an improvement in reason-

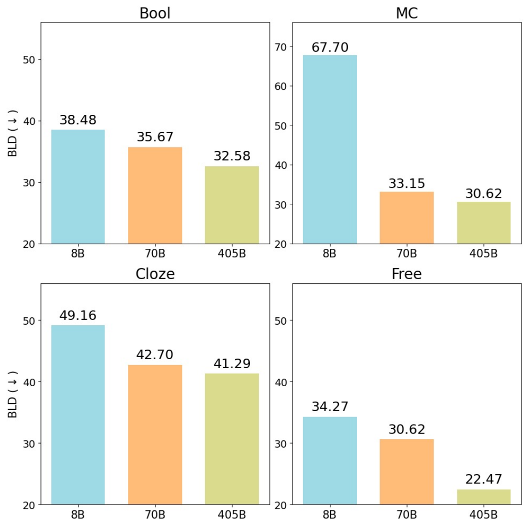

Figure 3: BLD performance comparison by the model size variation within the LLaMA3.1 family. The x-axis represents the parameter size (8B, 70B, and 405B).

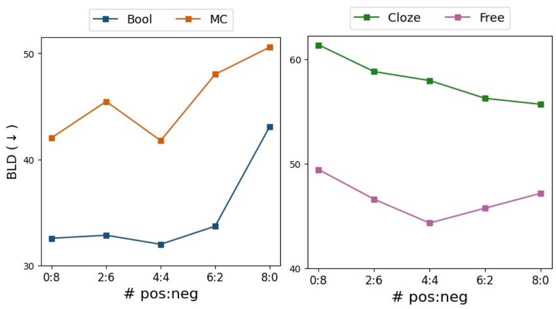

Figure 4: Performance changes of LLaMA3.1 according to the ratio configuration of polarity type (positive/negated) in ICL samples.

ing capabilities (Wei et al., 2022a; Touvron et al., 2023). However, these studies have primarily been conducted in scenarios where positive queries are presupposed. Therefore, we observe whether this improved reasoning ability is maintained even in negated scenarios. Figure 3 shows the BLD scores for each task according to the variation of parameter size within the LLaMA3.1 model family[6].

The results demonstrate that as the model's parameter size increases across all tasks, the BLD score decreases. In particular, in the MC task, the largest 405B model shows a significant alleviation in the BLD score by 37.08 points compared to the 8B model. Thus, the increase in model size normally leads to a reduction of *negation blindness*, although the extent of this effect varies depending

[6]For more results related to insight 2 (including the Qwen model and another series of LLaMA models), please see Appendix C.3.

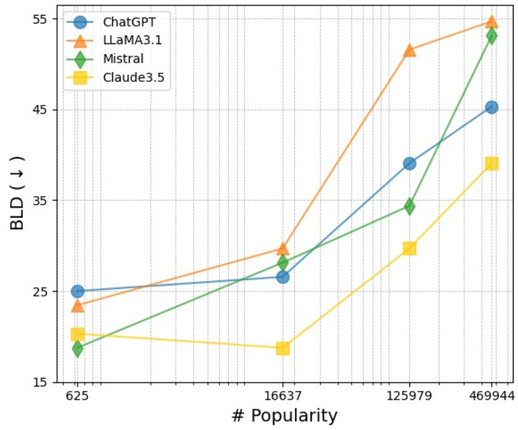

Figure 5: Performance changes of each model according to the entity's popularity. Each vertex of the x-axis represents a quartile in the distribution range of popularity.

on the task. This suggests the need for continued research on the *negation blindness* problem.

Insight 3: Excessive provision of positive exemplars in selection tasks and negated exemplars in generation tasks can deteriorate the problem. We explore the effect of the composition ratio of positive and negated exemplars on *negation blindness* in an in-context learning setting. In this study, the total number of shots is set to 8. Figure 4 shows the variation in BLD performance according to the composition ratio of different types of shots.

Different patterns exist between selection tasks (Bool and MC) and generation tasks (Cloze and Free). First, in the selection tasks (left plot chart), it is observed that the BLD score tends to be higher when the number of exemplars related to negated queries is smaller than that of exemplars related to positive queries. Specifically, when the number of positive and negated exemplars is 8:0 or 6:2, *negation blindness* occurs more frequently than the opposite cases (0:8 or 2:6 cases). Notably, when only positive exemplars are provided (8:0 case), the BLD score exhibits a sharp increase compared to the case where only negated exemplars are provided (0:8 case), with a worsening of 10.51 points in the Bool task and 8.52 points in the MC task.

On the other hand, the generation tasks (right plot chart) show the opposite tendency to the selection tasks. Shots consisting of only negated exemplars (0:8 case) show higher BLD scores than those with only positive exemplars (8:0 case). In particular, in the Cloze task, a BLD score increase of 5.68 points is observed, from 55.68 to 61.36.

In sum, from the perspective of the effective-

Model	Method	BLD (↓)				
		Bool	MC	Cloze	Free	Avg.
ChatGPT	Vanilla	30.90	35.96	49.16	42.98	39.75
	CoT	35.39	23.31	41.01	39.61	34.83
	Decom	35.11	33.43	50.84	44.66	41.01
	Refine	30.62	28.93	47.47	46.91	38.48
LLaMA3.1	Vanilla	38.48	67.70	49.16	34.27	47.40
	CoT	36.80	43.26	48.31	32.02	40.10
	Decom	30.34	63.20	53.09	30.06	44.17
	Refine	31.74	43.54	46.63	24.44	36.59
Mistral	Vanilla	31.74	67.13	42.70	27.81	42.35
	CoT	28.37	50.84	35.96	26.97	35.53
	Decom	25.00	49.16	37.36	26.12	34.41
	Refine	29.21	47.19	37.08	28.93	35.60
Claude3.5	Vanilla	49.72	33.99	53.37	26.12	40.80
	CoT	47.75	19.38	25.84	23.03	29.00
	Decom	52.81	19.10	43.26	25.42	35.15
	Refine	45.51	33.43	35.67	24.16	34.69

Table 2: Results across various self-enhancement methods for each model. Blue scores are cases where the BLD score worsened compared to the vanilla method.

ness of providing few-shot examples, the optimal ratio of negated and positive exemplars may vary depending on the type of task being performed.

Insight 4: Knowledge with higher popularity is more likely to undermine the model's logical coherence in handling negation. Figure 5 shows the changes in BLD performance of each model as the increase in popularity of the entity corresponding to the ground-truth answer y. LLMs commonly exhibit higher BLD scores for queries related to entities with high popularity, which indicates a worsening of the *negation blindness* problem. Notably, the difference in BLD scores is not significant up to the Q2 (16637) quartile of the popularity distribution, but it becomes even larger from the Q3 point (125979). For example, in the case of LLaMA3.1, the BLD score in Q1 and Q4 fluctuated the most, by 31.25 points.

Inspired by results demonstrated in previous studies, we interpret these findings as a knowledge shortcut phenomenon caused by a bias towards positive knowledge in the model's inherent knowledge distribution (Chen et al., 2023). In other words, due to the stronger confirmation bias of LLMs towards more popular knowledge, they may respond to a negated query in the same way as a positive query (Nickerson, 1998; Xie et al., 2023).

Model	Method	BLD (↓)				
		Bool	MC	Cloze	Free	Avg.
ChatGPT	Vanilla	30.90	35.96	49.16	42.98	39.75
LLaMA3.1		38.48	67.70	49.16	34.27	47.40
Claude3.5		49.72	33.99	53.37	26.12	40.80
Multi-agents	Voting	37.92	42.13	55.06	50.84	46.49
	Debate	**6.46**	**30.34**	**35.67**	**25.56**	**24.51**

Table 3: Effectiveness comparison of multi-agent methods and single models. **Bold** indicates the highest score.

5.2 Exploring Alleviation Feasibility with Reasoning Enhancement Approaches

This section explores the feasibility of alleviating the *negation blindness* problem with various reasoning enhancement methods.

We first examine methods that strengthen shallow reasoning (Insight 5) and deep reasoning (Insight 6), respectively. Subsequently, we extend the scope of these approaches from an individual, self-reasoning paradigm to a broader multi-agent setting, thereby exploring the potential of collaborative reasoning for mitigation (Insight 7).

Insight 5: Shallow reasoning can sometimes exacerbate logical incoherence in negation. Table 2 shows the results of applying various self-reasoning enhancement methods known to help improve the reasoning ability of LLMs via a prompt engineering approach. We explore the feasibility of mitigating *negation blindness* by adopting chain-of-thought (CoT) (Wei et al., 2022b), task decomposition (Decom) (Khot et al., 2022), and self-refine (Refine) (Madaan et al., 2024) methods.

Methods that enhance reasoning by adding intermediate reasoning steps or using self-reflection show that the *negation blindness* scores are alleviated in various tasks. For example, when these reasoning-enhanced methods are applied to the Mistral model, the BLD score is reduced compared to the vanilla method in all methods and all tasks. In particular, in the MC task, the self-refine method shows an alleviation of 19.94 points (67.13 $\rightarrow$ 47.19) compared to the vanilla method. The LLaMa3.1 and Claude3.5 models also alleviate all tasks except for the task decomposition method.

However, what we should pay attention to is the blue cases where the BLD performance worsens compared to the vanilla method that only includes task instructions. Notably, ChatGPT shows an increase in its BLD score in at least one task when applying each reasoning-enhanced method. For ex-

Metrics	Accuracy (↑)		BLD (↓)
	pos	neg	
Model	Free-form		
Vanilla-Qwen2.5-7B	88.76	49.72	46.91
TTS-Qwen2.5-7B	85.96	80.62	**17.42**

Table 4: Results for a trained model using the test time scaling (TTS) technique that enhances deep reasoning

ample, the task decomposition method shows a maximum BLD deterioration of 4.21 points (Bool task) in Bool, Cloze, and Free tasks.

Therefore, it suggests that methods that perform the reasoning in multi-steps by actively inducing the inherent knowledge of LLM may damage the logical coherence of the model for negation, depending on the model.

Therefore, these results indicate that efforts to actively induce the internal knowledge of LLMs through multi-step reasoning may, in certain cases, undermine the model's logical coherence with respect to negation, depending on the model architecture. Specifically, attempts to enhance shallow reasoning ability solely through task-instruction prompt engineering on general-purpose LLMs—without any additional training—appear limited in mitigating the *negation blindness*.

On the impact of reasoning chains and *negation blindness*, we attribute this issue to error propagation during the generation of reasoning chains, which ultimately leads to inappropriate final predictions[7]. This claim aligns with prior observations in other tasks, where cumulative errors in multi-step reasoning processes have been extensively reported (Mukherjee et al., 2025).

Insight 6: Deep reasoning enhances logical coherence under negation. Based on the empirical findings, we observe that models with deeper reasoning capabilities demonstrate improvements in BLD scores, indicating reduced *negation blindness*.

In particular, models that leverage deeper reasoning via test-time scaling (TTS) techniques (Snell et al., 2025; Liu et al., 2025; Muennighoff et al., 2025) show notable enhancements. Table 4 presents a comparison between the TTS-enhanced version of Qwen2.5 (built following Muennighoff et al. (2025)) and the vanilla Qwen2.5 model. The TTS-based trained Qwen model substantially reduces BLD scores by 29.49 points, improving them from 46.91 to 17.42[8]. For qualitative analyses of this, please refer to Appendix C.4.

Insight 7: Debate between multi-agents is effective in alleviation. To investigate the feasibility of mitigation, we explore approaches based on multi-agents in addition to the reasoning enhancement method by a single LLM. Table 3 shows the comparative performance between single models' responses and multi-agent-based methods' responses. The composition of the multi-agent technique consists of three single models (ChatGPT, LLaMA3.1, and Claude3.5) as presented. First, majority voting (Voting) is a relatively simple method that predicts the final answer by combining the responses of single models, but it often yields limited benefits or even negatively affects performance.

On the other hand, the multi-agent debate method, based on the multiple steps of each agent's opinion generation and mutual discussion process, shows its effectiveness by consistently improving performance in all tasks. For example, the debate method shows an alleviation effect of 43 points or more compared to Claude3.5 in the Bool task and 37 points or more compared to the LLaMA3.1 model in the MC task. This suggests that a single model's logical incoherence by negation can be effectively supplemented through mutual discussion.

6 Conclusion

We investigate the phenomenon wherein LLMs fail to recognize semantic inversion induced by negation, despite correctly encoding the corresponding affirmative knowledge. We define this systematic failure as *negation blindness* and present the first comprehensive study targeting this issue within the broader landscape of negation in LLMs.

To this end, we introduce a unified verification framework—encompassing task design and evaluation schema—designed to diagnose and mitigate *negation blindness*. Our framework supports rigorous empirical analyses and facilitates the exploration of alleviation strategies, such as self-enhancement techniques. Supplementary experiments confirm the framework's applicability across diverse contexts.

[7] Representative error cases, where CoT-based approaches fail to appropriately account for negation during intermediate reasoning steps, are provided in Appendix C.6.

[8] Table 10, which shows the performance differences between OpenAI's ChatGPT and the o1 model (which performs deeper inference), also supports this claim.

Limitations

Implications. By positioning *negation blindness* at the intersection of knowledge grounding and negation comprehension, this work advances our understanding of the logical consistency of LLMs under negated conditions. Furthermore, our framework generalizes beyond the immediate study, offering implications for causal, temporal, and commonsense reasoning, as well as high-stakes applications in law and medicine where nuanced linguistic shifts are critical. Collectively, our contributions offer a scalable foundation for diagnostic evaluation and optimization of LLM reasoning under negation.

Limitations. Our proposed verification framework serves as a basis for alleviation by quantifying the *negation blindness* problem in positive and negated queries based on QA situations, and hallucinations regarding contexts or entities are observed occasionally in the generated responses. However, since the issue of hallucinations is a severe problem even in large language models with enormous parameter sizes, it is required for our NLP communities to continue to solve the challenge.

While the proposed *BLD* score serves as a metric to evaluate *negation blindness*, it does not capture factors such as the model's training objective or neuron-level analysis within the network. Nonetheless, we believe that our findings lay a foundation for future research aimed at developing AI models that effectively address this issue.

Also, due to the issues of API cost and GPU resources for various LLMs, experiments were conducted with examples randomly sampled from the entire data, and not all cost-required models, including the o3 model (OpenAI), were adopted. A relatively larger number of study cases may be needed to assess the full spectrum of capabilities. API-based LLMs' generated results may vary depending on changes in the model version.

We plan to improve our framework for future work by conducting human evaluations with a considerable cases and enhancing the way of qualitative analysis for addressing the model's hallucinated answers. As miniaturization technology advances, verification of sLLMs with more compressed parameter sizes is also a desirable direction for GPU resource issues.

Ethics Statement

We discuss the main ethical considerations of the framework we proposed: (1) Privacy. the datasets adopted to experiment with our framework provide (factual information sourced from the web or Wikipedia.), and our verification results do not contain privacy issues. (2) Potential problems. Although we take conscientious steps to ensure the quality of our framework and resources, there can still be potential problems with the quality of the generated results, which can lead to incorrect predictions in applications that leverage factual information and negation mechanisms. (3) Model deployment. Our approach employs the pre-trained large language models (LLMs) for the downstream tasks, which have the risk of reflecting the bias of the training data. It is a well-known threat in tasks using PLMs and LLMs, and we should be careful about social impact when using this method since our approach aims to handle factual information and its negated content.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by Institute of Information communications Technology Planning & Evaluation(IITP) under the Leading Generative AI Human Resources Development(IITP-2025-R2408111) grant funded by the Korea government(MSIT). This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2025-RS-2020-II201819). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence star fellowship support program to nurture the best talents (IITP-2025-RS-2025-02304828) grant funded by the Korea government(MSIT).

References

AI Anthropic. 2024. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3:6.

Hiba Arnaout and Simon Razniewski. 2023. Can large language models generate salient negative statements? *arXiv preprint arXiv:2305.16755*.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2022. Uncommonsense: Informative negative knowledge about everyday concepts. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 37–46.

Nicholas Asher and Swarnadeep Bhar. 2024. Strong hallucinations from negation and how to fix them. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12670–12687, Bangkok, Thailand. Association for Computational Linguistics.

Stephen Barker and Mark Jago. 2012. Being positive about negative facts. *Philosophy and Phenomenological research*, pages 117–138.

Marianne Celce-Murcia, Diane Larsen-Freeman, and Howard Alan Williams. 1983. *The grammar book: An ESL/EFL teacher's course*. Newbury House Rowley, MA.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9890–9908, Toronto, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of plms' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. *arXiv preprint arXiv:2105.03519*.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer learning for natural language processing workshop*, pages 52–62. PMLR.

Myeongjun Jang, Frank Mtumbuka, and Thomas Lukasiewicz. 2022. Beyond distributional hypothesis: Let language models learn meaning-text correspondence. *arXiv preprint arXiv:2205.03815*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

DR Krathwohl. 2002. A revision bloom's taxonomy: An overview. *Theory into Practice*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*.

Charles R MacDonald. 1965. The role of negation in human knowledge. *Laval théologique et philosophique*, 21(1):80–114.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Marvin Minsky. 1997. Negative expertise.

Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope: Guidelines v1. *Computational linguistics and psycholinguistics technical report series, CTRS-003*, pages 1–42.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*.

Sagnik Mukherjee, Abhinav Chinta, Takyoung Kim, Tarun Anoop Sharma, and Dilek Hakkani Tur. 2025. Premise-augmented reasoning chains improve error identification in math reasoning with llms. In *Forty-second International Conference on Machine Learning*.

Raymond S Nickerson. 1998. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175–220.

OpenAI-Blog. 2022. Chatgpt: Optimizing language models for dialogue.

Amit Rozner, Barak Battash, Lior Wolf, and Ofir Lindenbaum. 2024. Knowledge editing in language models via adapted direct preference optimization. *arXiv preprint arXiv:2406.09920*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. *arXiv preprint arXiv:2306.08189*.

Neeraj Varshney, Satyam Raj, Venkatesh Mishra, Agneet Chatterjee, Ritika Sarkar, Amir Saeidi, and Chitta Baral. 2024. Investigating and addressing hallucinations of llms in tasks involving negation. *arXiv preprint arXiv:2406.05494*.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

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

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

A. Yang et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

A Demonstration of Semantic Validity in Negated Scenarios

In this section, we investigate whether the logical theory of distributional inversion under negation—particularly as formulated by Asher and Bhar (2024)—can be extended to generative LLMs and their associated tasks. This theoretical foundation serves as the basis of our study. To this end, we provide both principled justifications for our proposed verification methodology and complementary empirical evidence.

A.1 Applicability Across Model Architectures

The phenomenon of *negation blindness* is not contingent on model architecture but arises from the statistical nature of language modeling itself. Since all language models define string-based probability distributions, they share foundational vulnerabilities irrespective of being encoder-based (e.g., BERT-type) or auto-regressive (e.g., GPT-type). Asher and Bhar (2024) formally articulate this universality: for any string-based probability distribution μ, the constraint $\mu(\phi) + \mu(\neg\phi) < 1$ inevitably violates probability axioms, preventing models from consistently assigning logical truths. Consequently, architectural differences do not invalidate the theoretical basis of BLD scoring, as the underlying limitations remain invariant across model classes. While empirical analyses in Asher and Bhar (2024) are limited to classification tasks, their theoretical proofs extend directly to generative models, where errors become even more pronounced under extended reasoning chains.

A.2 Applicability Across Task Types

The theoretical assumptions underlying BLD scoring apply equivalently to both classification and generation tasks. Masked language models and auto-regressive models share the same learning objective—minimization of conditional cross-entropy losses over $log\ p\theta(x_t|\cdot)$—with differences confined to implementation details such as masking and caching. Asher and Bhar (2024)'s proofs are grounded in Transformer properties (e.g., attention locality, finite context windows) and conditional probability prediction, not in the presence of masks, making the classification–generation distinction theoretically immaterial. Moreover, tasks such as yes/no QA or NLI classification can be reformulated as generation tasks, while auto-regressive models can perform classification by comparing log-probabilities of candidate outputs.

A.3 Bridging Theory and Empirical Validation

Our study leverages this theoretical equivalence to extend Asher and Bhar (2024)'s framework to large-scale auto-regressive LLMs (>7B parameters), empirically validating the persistence of BLD scoring in generative settings. This demonstrates that the phenomenon is not confined to a specific task formulation but represents a universal limitation of statistical language modeling. The alignment between theoretical predictions and empirical outcomes underscores the robustness of BLD scoring as a generalizable lens for analyzing logical coherence in LMs.

B Experimental Details

B.1 Raw Dataset License and Statistics

# QA pairs (total)	95,956
# unique answers	40,478
# evidence documents	662,659
Avg. question length (word)	14
Avg. document length (word)	2,895

Table 5: Data statistics of TriviaQA.

TriviaQA (Joshi et al., 2017) dataset has an Apache 2.0 license. This license may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that this license meets the following conditions: this license must give any other recipients of the Work or Derivative Works a copy of this License; and this license must cause any modified files to carry prominent notices stating that this license changed the files; and this license must retain, in the Source form of any Derivative Works that this license distributes, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works.

Table 5 shows the statistics of the TriviaQA dataset[9]. In particular, this study adopts the web version of the data splits as the raw dataset to utilize

[9] In particular, since some examples contain exceptionally long input contexts, we selected only models supporting an input context length of at least 16K tokens.

the web-crawled context data, which better reflects real-world environments.

B.2 Tools

One of the widely used English-based natural language understanding (NLU) tools, spaCy[10], was employed as the named entity recognition (NER) model for constructing the option set in the multiple-choice selection task (Algorithm 1).

Also, the popularity of the entities analyzed in Figure 5 was measured using Wikipedia page views, following the method of work (Mallen et al., 2023) of the popQA dataset, and these can be obtained by calling the Wikipedia API. Please note that since we used examples from the web version of the dataset to set up a verification experiment close to the real-world environment, there are cases where the entities used in the ground-truth answers do not overlap with the entities in the corresponding Wikipedia pages.

B.3 Error Cases during the Verification Dataset Construction Process

Here, we present qualitative examples of error cases that occurred while constructing a dataset for verification of *negation blindness*.

First, during the negation process of initial queries, grammatical errors are often observed, including incorrect use of auxiliary verbs or tense mismatches. For example, the positive query "Which musical featured the song The Street Where You Live?" $\rightarrow$ the negated query "Which musical doesn't feature the song The Street Where You Live?" To minimize such errors, we adopted a few-shot prompting approach, using diverse negation cases as guidelines. For instance, negating "Who set the fire $(\cdots)$" to "Who did not set the fire $(\cdots)$" requires not only inserting a "not" token but also ensuring correct tense and auxiliary verb usage. We provided detailed exemplars covering these nuanced cases to guide the construction process.

Furthermore, there are error cases in the option set construction (for the MCQA task). The option set construction was designed with a rule-based algorithm (Alg.1) to ensure that distractor options remained within the same entity type scope as the target answer. This design increases task difficulty for QA models by selecting distractors with the same semantic type rather than generating options arbitrarily via random sampling or generative models.

Context "Augustus: Birth and Inheritance of Augustus' many names and honorifics, historians favor three of them, each for a different phase in the emperor's life. From his birth in 63 B.C. he was Octavius; after his adoption was announced in 44 B.C., $(\cdots)$"
Question "Who was the first emperor of Rome?"
Extracted Choices {"text": ["velletri", "cleopatra", "mark antony", "gaivs•ivlivs•caesar•octavianvs"], $(\cdots)$}

Table 6: Error cases during the option construction process

A typical error case in this process can be found in Table 6. As shown in the table, "Velletri" (an Italian city which should be tagged as `GPE`, Geo-Political Entity) was incorrectly tagged as `PERSON`. Such misclassifications arise from the adopted entity extraction tool, SpaCy. Nonetheless, all option sets underwent thorough human-based revision after automated processing to correct such entity type misclassifications.

B.4 Hyperparameters

For the verification experiments, the temperature is set to 0.75, top-P is set to 0.9, and the maximum output length is set to 1024. Also, in the few-shot settings (Figure 4) to verify the effectiveness of in-context learning, the number of shots is set to 8.

For the other hyperparameter settings, we follow the recommended guidelines provided by each model's provider, such as OpenAI and Meta.

In particular, it is important to note that API-based models might occasionally generate empty responses due to network transmission timeouts or API overload. In such cases, we followed the standard practice of resubmitting the request until obtaining non-empty responses.

We should emphasize that to prevent any potential influence from prior responses, we cleared the input history each time we submitted a new query to the API-based models. Unless otherwise specified, we refrained from engaging in any further attempts with API-based models to modify their responses.

[10] https://github.com/explosion/spaCy

B.5 Textual Entailment-based Evaluation Mechanism

The *negation blindness* is quantified through the formalization of the BLD score (§ 3.2). In detail, given a model's predicted answer and the corresponding ground-truth answer for a query, the LLM evaluator was used to classify the semantic entailment between the two texts as either "`[[entailment]]`" or "`[[contradiction]]`". The same model used for negation in data construction, GPT-4o-mini (version gpt-4o-mini-2024-07-18), was employed as the evaluator. The prompt template can be found in Table 32 below.

B.6 Description of Reasoning Enhancement Methods for Mitigating *Negation Blindness*

B.6.1 Self-enhancement

Reasoning-enhancement prompt engineering methods, which actively leverage the parametric knowledge learned through pretraining in LLMs, have demonstrated improved performance in tasks requiring high levels of inference. Among these, the methodologies utilized in our mitigation experiments (Table 2) are as follows.

Chain-of-thought. Zero-shot chain-of-thought (CoT) is a reasoning approach where a model generates a sequence of intermediate thoughts or reasoning steps to solve a problem without any prior examples or training specific to that task. It's designed to enhance the model's ability to tackle complex tasks with intermediate logical steps, even when encountering the problem for the first time (Wei et al., 2022b).

Task Decomposition. Task decomposition in prompt engineering involves breaking down a complex task into smaller, more manageable sub-tasks. By addressing each sub-task individually, it becomes easier to guide a language model to achieve the desired outcome. This approach helps in clarifying requirements, reducing ambiguity, and improving the overall effectiveness of the prompts (Khot et al., 2022).

Self-refine. Self-refine methodology in prompt engineering involves iteratively improving the generated results by using the model's own feedback. The process includes generating responses from the initial prompt, analyzing these outputs to identify flaws or areas for improvement, and then refining the results accordingly to enhance performance and accuracy (Madaan et al., 2024).

B.6.2 Multi-agent based Enhancement

Multi-agent-based approaches have recently attracted attention for their strong performance capabilities (Liang et al., 2023; Wang et al., 2024). The methods evaluated in Table 3 are as follows.

Majority Voting. Multiple models[11] generate responses for a given task, and the response selected by the majority is adopted as the final prediction. If no response is selected by the majority, the response from a randomly chosen model is adopted.

Multi-agent Debate. The debate process involves the following steps: First, each model generates an initial response to the given task. Then, these responses are collected and presented as anonymized individual opinions. Each agent subsequently formulates arguments that include their agreement/disagreement with these opinions. Finally, the moderator model mediates opinions and generates a comprehensive judgment based on the entire discussion process, which is used as the final prediction.

B.7 Model Training Detail

To further investigate the potential of test-time scaling (TTS) methods for enabling deeper reasoning, we conducted additional experiments (Appendix C.4) where we trained models by adopting the methodology introduced in Muennighoff et al. (2025).

Following Muennighoff et al. (2025), we conducted supervised fine-tuning (SFT), which was identified as the key driver of performance gains. The training was carried out using eight NVIDIA A100 GPUs for approximately 35 minutes, with 5 epochs, a batch size of 16, and a learning rate of 1e-5 with cosine decay. We utilized the publicly available s1K dataset, which consists of 1,000 high-quality reasoning samples.

C Further Analysis

C.1 Additional Results on the Extension of Source Datasets

Regarding the range of datasets used as raw sources for verification, we have conducted additional experiments to further demonstrate the scalability

[11] In this study, ChatGPT, LLaMA 3.1, and Claude 3.5 were utilized.

Metrics	Accuracy (↑)		BLD (↓)
NQ dataset	pos	neg	
Model	Free-form		
ChatGPT	83.52	45.05	46.15
LLaMA3.1	91.21	54.95	40.66
Mistral	86.81	58.24	37.36
Claude3.5	90.11	61.54	**36.26**
Model	Cloze-style		
ChatGPT	79.12	52.75	43.96
LLaMA3.1	78.02	57.14	36.26
Mistral	87.91	67.03	**29.67**
Claude3.5	84.62	48.35	46.15

Table 7: Verification experiment results for the NQ dataset

and generalizability of our proposed verification methodology.

Specifically, we constructed an additional verification set based on the widely utilized Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), in addition to TriviaQA. This new verification set was created following the same procedure detailed in Section 3.4, and the models evaluated remained identical to those employed in our primary experiments (Table 1). For clarity, we note that the NQ dataset contexts were curated by filtering for examples with original context lengths exceeding 16K tokens. This preprocessing step involved removing noise, such as special symbols and escape characters arising from HTML parsing, to ensure the inclusion of sufficiently long contexts for QA task difficulty and prevent overly short context examples.

Table 7 shows the experimental results of the same models on the NQ dataset. First, the results in free-form generation exhibit trends consistent with those reported in Table 1. For example, the Claude model achieves the lowest BLD score (36.26), indicating the least degree of negation blindness, whereas ChatGPT records the highest BLD score (46.15).

Results from the cloze-style completion task also tend to align with the findings observed in Insight 1. For instance, the Mistral model achieves the highest accuracy (67.03) on negated queries and demonstrates the lowest degree of negation blindness (BLD score of 29.67) among evaluated models, indicating superior performance in handling negation.

These results provide supporting evidence for the extensibility of our verification methodology across diverse datasets, thereby reinforcing its broader applicability and robustness.

C.2 Further Experiments for Enhancing the Diversity of Negation Constructions

To construct a richer set of negation scenarios, we conducted additional experiments that utilize randomly selected negation cues based on a comprehensive list encompassing diverse forms of negation. Specifically, we created a set of negation expressions including *{not, no, never, none, no one, nobody, nothing, nowhere, neither, nor, hardly, barely, scarcely, rarely, seldom, no longer, not at all, by no means, not any}*. For data construction, we randomly selected negation cues from this set to generate negated instances. Apart from this step, all other data construction procedures remain identical to those described in Section 3.4.

However, constraints are imposed to ensure that the semantic intent of the initial query was not compromised. For example, in queries such as "Who was born first, Kiefer Sutherland or Christian Slater?", it is inappropriate to use components such as *"nowhere"*. Instead, expressions like *not, never, neither, or none* enable proper semantic inversion. For instance, this newly built dataset in this process (from NQ dataset) includes sentences with semantic inversion induced by randomly selected negation cues, such as:

- "Who *hardly* smokes the hookah in Alice in Wonderland?"
- "What was *by no means* Blondie's last UK No. 1 of the 80s?"
- "Which Scottish newspaper does *never* feature the Broons and Oor Wullie?"

Metrics	Accuracy (↑)		BLD (↓)
NQ w/div. negation	pos	neg	
Model	Free-form		
ChatGPT	**89.33**	51.65	45.05
LLaMA3.1	87.92	52.75	46.15
Mistral	85.67	**61.54**	**36.26**
Claude3.5	84.83	60.44	37.36

Table 8: Verification results on the NQ dataset constructed using diverse negation cues

The experimental results from these approaches, which guarantee the diversity of negation cues, are presented in Table 8. These results align with the general trends observed in insight 1 (§ 5.1. In particular, according to the BLD scores, the Mistral and Claude 3.5 models exhibit substantially greater robustness to *negation blindness* problem (36.26 and 37.36, respectively) compared to ChatGPT and LLaMA 3.1 (45.05 and 46.15, respectively).

These results further strengthen the robustness and credibility of our experimental setups, supporting the comprehensiveness of our analyses on negation.

C.3 Additional Results on Model Size Scaling

Metrics	Accuracy (↑)		BLD (↓)
	pos	neg	
Model	Free-form		
LLaMA3.2-1B	65.17	65.73	46.07
LLaMA3.2-3B	87.08	55.06	42.7
LLaMA3.2-11B	88.76	55.90	**41.01**
Qwen2.5-3B	83.71	50.00	47.47
Qwen2.5-7B	88.76	49.72	46.91
Qwen2.5-32B	85.39	53.93	42.98
Qwen2.5-72B	91.85	60.11	**38.20**

Table 9: Results of changes in accuracy and BLD scores according to changes in model size: Qwen2.5 and LLaMA3.2

To examine whether the findings reported in Insight 2 of Section 5.1 for the LLaMA3.1 model series extend to other architectures, we conducted the same set of experiments on additional model families, namely Qwen2.5(Yang et al., 2024) and LLaMA3.2(Dubey et al., 2024).

As shown in Table 9, the results are consistent with those observed for the LLaMA3.1 series. Specifically, as the parameter size of the models increases, the BLD score gradually decreases, indicating that the degree of logical coherence disruption caused by the phenomenon of *negation blindness* is reduced.

C.4 Additional Performance Results from More Powerful Reasoning Models

Adoption of the o1 model. In the main text, we already presented findings on the relationship between model capacity and the degree of *negation blindness* across the LLaMA family (Figure 3 and Insight 2). Building on that analysis, we conducted an additional experiment involving a more powerful reasoning-based model, OpenAI's o1 model. In Table 10, the performance of the o1 model is compared against ChatGPT on the free-form generation task.

Model	Accuracy			BLD (↓)
	pos (↑)	neg (↑)	avg (↑)	
ChatGPT	89.33	55.34	72.33	42.98
o1	**91.29**	**62.92**	**77.11**	**35.11**

Table 10: Evaluation results of comparing the o1 model with ChatGPT on the free-form generation task.

Context
`{{context text}}`

Positive Query
"Gjetost is the national cheese of which country?"
–
o1's Response: "Norway."
ChatGPT's Response: "Norway."

Negated Question
"Gjetost isn't the national cheese of which country?"
–
o1's Response: "United States."
ChatGPT's Response: "Norway."

Table 11: Qualitative example #1 of the generation results of o1 and ChatGPT

Context
`{{context text}}`

Positive Query
"What is the capital of the U.S. state of Connecticut?"
–
o1's Response: "Hartford."
ChatGPT's Response: "Hartford."

Negated Question
"What isn't the capital of the U.S. state of Connecticut?"
–
o1's Response: "Des Moines is not the capital of Connecticut."
ChatGPT's Response: "Hartford."

Table 12: Qualitative example #2 of the generation results of o1 and ChatGPT

The results show the observations aligning with the results discussed in Section 5.1, where we found meaningful improvements in negation handling as model capacity increased within the

LLaMA family. In other words, the o1 model demonstrates a significantly lower degree of *negation blindness* compared to ChatGPT, showing a difference of about 7 points or more. Moreover, Table 11-12 are qualitative examples that illustrate this tendency well.

Adoption of the TTS method. As presented in Insight 6 of Section 5.1, we provided quantitative evidence that deep reasoning substantially enhances the logical coherence of LLMs under negation. Specifically, by employing test-time scaling techniques that enable deeper reasoning during inference, we demonstrated a significant improvement in the BLD score.

In addition to these quantitative findings, we further support this claim with qualitative examples presented in Table 13. These examples illustrate the mechanisms through which deep reasoning contributes to improved coherence, and we provide a detailed analysis of their implications.

First, as observed in Example #1, the reasoning-enhanced model trained with TTS accurately detects the semantic inversion introduced by negation in queries. This aspect is particularly evident in its generated reasoning paths. Even if negation is not captured in the initial reasoning steps, subsequent reasoning stages successfully identify the negation, thereby preserving logical coherence in their responses. (e.g., "But wait, the question is asking which player was NOT sent off.", "So if Keane was sent off, then the answer would be someone who wasn't.") This trend is also consistent with the o1's response examples presented in Tables 11 and 12.

Next, according to Example #2, the Qwen model trained with TTS demonstrates its ability to recognize the negation components within the query. Based on this recognition, it reverses the distribution of the original positive query answer accordingly. For instance, the model produces reasoning such as, "The question is asking which three aren't the most expensive." and "So, I need to identify the top three and then pick options that aren't among them."

These examples illustrate cases in which the model correctly performs the "inversion of the answer distribution induced by negation", which is precisely what our BLD score aims to measure.

Thus, these results suggest that the observed relationship between deep reasoning ability and negation detection substantiates that the negation blindness exhibited by vanilla models is not merely due to treating negation components as trivial tokens or typographical noise.

Interpretation of Negation Blindness in Vanilla Models Without Deep Reasoning. As discussed in Figure 5 and Insight 4 (§ 5.1), we observe a correlation between the popularity of entities in queries and their associated BLD scores, where entities frequently encountered during (pre-)training lead to higher degrees of *negation blindness*.

We interpret this as follows: vanilla models with limited reasoning capability tend to form semantic shortcuts (Chen et al., 2023) for popular entities. Given the substantial skew in training data distributions—where positive form data vastly outnumbers negated or negative form data—such shortcuts emerge almost inevitably, except in a handful of curated benchmarks.

Consequently, models with stronger reasoning capacity rely less on knowledge shortcuts and instead maintain logical coherence by engaging in deeper inference processes, thus better recognizing and handling negation.

C.5 Additional LLM-as-a-Judge Evaluation of Self-enhancement Methods

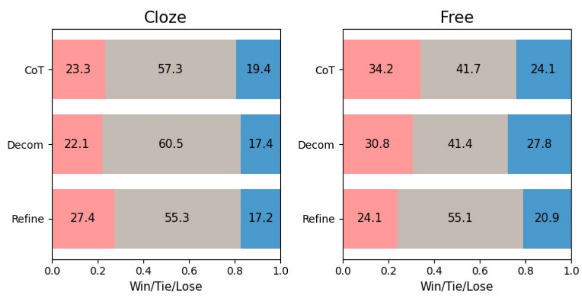

Figure 6: LLM-as-a-Judge evaluation results for Mistral. In order to reduce position bias, the actual experiment was performed twice by switching A and B.

Considering that LLMs' responses can vary in low-constrained generation tasks, we conduct an evaluation using the LLM-as-a-Judge method on both Cloze and Free tasks. Specifically, we conducted an evaluation using a pair-wise comparison method to compare responses generated by the vanilla approach and the reasoning-enhancement approach. To reduce the position bias of the evaluation, we performed evaluations twice by swapping the positions.

Figure 6 shows the LLM-as-a-Judge evaluation results. We conducted an evaluation on Mistral,

Example #1

Context
“This was to be one of the most excited Charity Shield matches ever witnessed. As early as the 8th minute, Eric Cantona played the ball back to Mark Hughes who scored one of his trade mark volleys. At the other end, Ian Wright $(\cdots)$"

Negated Question
"In the year 2000 Charity Shield match, which player was not sent off?"

1. TTS-Qwen2.5-7B
*** Reasoning paths:** "think Okay, let me try to figure out the answer. The question is asking about the 2000 FA Charity Shield match between Chelsea and Manchester United. Hmm. Let me recall... So, I remember that in the 2000 Charity Shield, Chelsea beat Manchester United 2-0. I think there was a sending-off, right? Wait, maybe Roy Keane got sent off. But was it in this particular match? Let me check my notes. Looking back at the context provided, there’s a section about the 2000 FA Charity Shield - Chelsea vs Manchester United. It says: Sadly for the Men from Manchester, the Boys in Blue of Chelsea got one over them. The Londoners won the 2000 Charity Shield by winning two goals to nil. Hasselbaink open his and Chelsea’s account with a goal after 22 minutes after beating Johnson and Stam. The second half got no better for the Reds as Roy Keane got sent off for a challenge on Gus Poyet. M Melchiot in the 73rd minute sealed the game for Chelsea as he scored past Bartez from the edge of the box. So according to this, Roy Keane was sent off in that match. But wait, the question is asking which player was NOT sent off. So if Keane was sent off, then the answer would be someone who wasn’t. Looking back at the match report: The second half got no better for the Reds as Roy Keane got sent off for a challenge on Gus Poyet. That suggests only Keane was sent off. So the players not sent off would be Chelsea’s players, like Hasselbaink, Melchiot, and perhaps others."
*** Prediction:** Hasselbaink

2. Vanilla-Qwen2.5-7B
*** Prediction:** Roy Keane

Example #2

Context
“Saffron - The most expensive Spice in the world - Saffron from Iran, the best in the world, www.saffron.name1 teaspoon ground saffron dissolved $(\cdots)$"

Negated Question
"What aren’t the world’s three most expensive spices?"

1. TTS-Qwen2.5-7B
*** Reasoning paths:** "think Okay, so I need to figure out which three spices aren’t the most expensive in the world. First, I remember that saffron is one of the most expensive spices because it’s so labor-intensive to make. The process of picking each strand by hand must drive up the cost. Then there’s vanilla; I think it’s also pretty pricey, though maybe not as much as saffron. But wait, are there spices that are even more expensive? Like black truffles or caviar—maybe they’re considered more luxurious than spices? Hmm. Let me think again. The question is asking which three aren’t the most expensive. So, I need to identify the top three and then pick options that aren’t among them. For example, if saffron, vanilla, and truffles are the top three, then any spice not on that list would be the answer. But which are actually the most expensive? answer The world’s three most expensive spices are saffron, vanilla, and safflower. Saffron is renowned for its intense labor requirements, making it the priciest. Vanilla follows due to its cultivation challenges, particularly for the Madagascar variety. Safflower, though less known, is also extremely costly because of the meticulous harvesting process where only the stigmas are collected. Other expensive spices like cardamom or star anise, while premium, don’t reach the top three. Thus, any spice not listed here (e.g., turmeric, cumin) would not be among the world’s three most expensive.
*** Prediction:** Turmeric, cumin, cinnamon

2. Vanilla-Qwen2.5-7B
*** Prediction:** Saffran

Table 13: Improved qualitative result examples when applying the test time scaling (TTS) method, which enables deeper reasoning during inference

which exhibited the most consistent performance mitigation in the mitigation experiment in Table 2. In both tasks, reasoning-enhancement methodologies achieved a higher win ratio compared to the lose ratio. For example, in the Cloze task, the self-refine method showed the highest win ratio of 27.4%, and in the Free task, the CoT method showed the highest win ratio of 34.2%. This suggests that the LLM-as-a-Judge evaluation results for applying the reasoning-enhancement method align with the results presented in Table 2.

C.6 Error cases of Chain-of-Thought method from the perspective of *negation blindness*

Our findings reveal that methods leveraging the intrinsic knowledge of LLMs—such as Chain-of-Thought (CoT)-based multi-step reasoning—can, depending on the model, degrade logical consistency in the presence of negation (see Table 2 and Insight 5). To substantiate this claim, we provide qualitative examples from our experiments.

Table 14 shows actual reasoning paths generated by the ChatGPT model for negated queries in the cloze-style completion task.

In Example 1, the reasoning chain produced by ChatGPT demonstrates that the model failed to capture the semantic inversion induced by negation in the query. Consequently, none of the intermediate reasoning steps attempted to address the negation, leading to an incorrect final prediction.

In Example 2, although ChatGPT's second reasoning path explicitly recognized the negation components in the query, it subsequently produced an overgeneralized inference in path 3 to maintain consistency with path 1, ultimately resulting in an incorrect final prediction.

These examples illustrate that for self-reasoning-enhanced methods such as CoT, error propagation across intermediate steps can critically impair the accuracy of the final output.

C.7 Additional Analysis on the Internal Mechanism of *Negation Blindness*

To gain further insight into the internal factors contributing to the observed *negation blindness* phenomenon, we conducted an attention-based analysis on open-source models such as Mistral and LLaMA. Specifically, we examined the attention scores assigned to negation components (e.g., phrases containing "not") across different tasks.

Table 15-16 show qualitative examples from

Example #1
Context "(...) With the possible exception of "Wild Horses," no Rolling Stones ballad is more beloved than "Angie." The mournful lyrics speak clearly of the sadness of love lost, which is strange coming from a guy who tends to sing songs like "Under My Thumb," where $(\cdots)$"
Negated Query "The surname of the woman who inspired the Rolling Stones song "Angie" is not [MASK]."
Reasoning Paths "Step 1: Mick Jagger and David Bowie were allegedly involved in a scandal with a woman named Angela. Step 2: The woman who claimed to be the inspiration for the Rolling Stones song "Angie" was David Bowie's ex-wife. Step 3: Based on the context, the surname of the woman who inspired the Rolling Stones song "Angie" is likely Bowie. [MASK]: Bowie"
Example #2
Context "(...) Ingrid Bergman won her third Oscar for her missionary looking after little brown babies - I thought she was a highlight indeed but in my modest opinion, Valentina Cortese for "Day For Night" deserved $(\cdots)$"
Negated Query "The Swedish actress [MASK] didn't win the Best Supporting Actress Oscar for Murder on the Orient Express."
Reasoning Paths "Let's think step-by-step. 1. The Swedish actress mentioned in the text who won the Best Supporting Actress Oscar for her role in "Murder on the Orient Express" is Ingrid Bergman. 2. The question states that this Swedish actress did not win the Best Supporting Actress Oscar for the movie. 3. Therefore, the correct answer to the question is Ingrid Bergman."

Table 14: Observed error cases arising from error propagation during the reasoning path generation process, when applying Chain-of-Thought (CoT) prompting

both Mistral and LLaMA3.1, derived from identical instances across tasks.

Type	q_{neg}	Attention Score
	Selection Task	
q^{bool}	Do {not twelve avenues} radiate from the Arc de Triomphe?	0.0545
q^{mc}	How many avenues do not radiate from the Arc de Triomphe?	0.0226
	Generation Task	
q^{cloze}	Simple Simon did not meet a [MASK] on his way to the fair.	0.0499
q^{free}	Who did not Simple Simon meet on his way to the fair?	0.1199

Table 15: Qualitative examples of potential internal mechanisms with Mistral.

Type	q_{neg}	Attention Score
	Selection Task	
q^{bool}	Do {not twelve avenues} radiate from the Arc de Triomphe?	0.045
q^{mc}	How many avenues do not radiate from the Arc de Triomphe?	0.0347
	Generation Task	
q^{cloze}	Simple Simon did not meet a [MASK] on his way to the fair.	0.0414
q^{free}	Who did not Simple Simon meet on his way to the fair?	0.0644

Table 16: Qualitative examples of potential internal mechanisms with LLaMA3.1.

The tendency of BLD scores reported in Table 1 of the main body suggests that negation blindness follows the trend: Free $\rightarrow$ Bool $\rightarrow$ Cloze $\rightarrow$ MC (least $\rightarrow$ most severe). As shown in the results of Table 15-16, the attention scores almost mirror that trend and may serve as indicative evidence for one potential internal factor contributing to *negation blindness*. For instance, within selection tasks, the attention score for negation components is higher in the Bool task than in the MC task. Similarly, in generation tasks, the Free task shows substantially higher attention to negation components compared to the Cloze task.

These additional findings hint at how variations in internal attention dynamics may influence the model's susceptibility to *negation blindness*.

C.8 Qualitative Analysis

Table 17-19 show model-specific qualitative examples for some tasks to verify *negation blindness*. In the multiple-choice selection task, LLaMA3.1 and Mistral failed to recognize negated queries and generated the same response as positive queries. In the boolean selection and cloze-style completion tasks, *negation blindness* cases are observed in LLaMA3.1 and Claude3.5.

As shown in Figure 3, the validation of *negation blindness* was conducted across different parameter sizes of the LLaMA3.1 family. Table 20 and 21 provide qualitative examples related to this. The 8B and 70B models exhibited *negation blindness* in both the multiple-choice selection task and the free-form generation task. However, the 405B model accurately captured the meaning reversal induced by both positive queries and negations, providing correct responses to both queries.

Furthermore, to provide additional insights, we present qualitative examples of the generation results for the self-reasoning enhancement methodologies explored in Section 5.2, which are presented in Table 22 and 23.

Context:
... After impressing a Capitol Records executive with his already elaborate live show, he was signed to a multi-album deal, the first of which was a revamped version of Feel My Power retitled Let's Get It Started. Producing an RB hit in "Turn This Mutha Out," Let's Get It Started went double platinum. Still, nothing could have foreshadowed the phenomenon of Please Hammer Don't Hurt 'Em, the 1990-released follow-up. Its first single, "U Can't Touch This," blatantly copped most of its hooks from Rick James ' funk classic "Super Freak," yet Hammer's added catch phrases (and young listeners' unfamiliarity with the original song) helped make it a smash. "U Can't Touch This" dominated radio and MTV during 1990 in a way few rap singles ever had, and won two Grammys (Best RB Song, Best Solo Rap Performance); save for a quirk in its release format – it was only available as a 12", which cut down on its sales – it would easily have been the first rap single to top the Billboard pop chart. The next two singles, "Have You Seen Her" (a flat-out cover of the Chi-Lites ' '70s soul ballad) and "Pray" (built on the keyboard hook from Prince's "When Doves Cry"), followed "U Can't Touch This" into the Top Ten, eventually pushing sales of Please Hammer Don't Hurt 'Em past the ten-million mark and making it the number one album of the year. Still, a backlash was growing against Hammer's frequent borrowing (some said theft) of classic hooks for his own hits; hip-hop purists also railed about his often simplistic, repetitive lyrics (indeed, "Pray" set a new record for the number of times its title was repeated during the song, at well over 100). $(\cdots)$ While it sold very well (over three million copies) and produced a sizable hit in the title track, Hammer's stage show had become as lavish as his lifestyle; loaded with singers, dancers, and backup musicians, the supporting concert tour was too expensive for the album's sales to finance, and it was canceled partway through.

Positive Query: Did MC Hammer have a hit record in 1990 with "U Can't Touch This"?

Negated Query: Did **{not** MC Hammer**}** have a hit record in 1990 with "U Can't Touch This"?

GT Answer: Yes. "It is because 'U Can't Touch This' was a major hit single released in 1990, dominating radio and MTV, and it won two Grammy Awards, confirming its success and impact on the music industry."

ChatGPT Prediction
positive
Yes. MC Hammer had a hit record in 1990 with "U Can't Touch This," which was the first single from his album "Please Hammer Don't Hurt 'Em." The song was a huge success, dominating radio and MTV, winning two Grammys, and contributing to the album becoming the best-selling rap album of all time. (✓)

negated
No. 'The question is asking whether someone other than MC Hammer had a hit record in 1990 with "U Can't Touch This." Since the question specifies "not MC Hammer,". (✓)

LLaMA3.1 Prediction
positive
Yes. The text states that "U Can't Touch This" dominated radio and MTV during 1990, won two Grammys (Best RB Song, Best Solo Rap Performance), and was a huge commercial success, pushing the sales of Please Hammer Don't Hurt 'Em past the ten-million mark.' (✓),

negated
Yes. MC Hammer did have a hit record in 1990 with "U Can't Touch This", which was a single from his album "Please Hammer Don't Hurt 'Em" and became a massive commercial success, topping the charts and winning two Grammys.'" (✗)

Mistral Prediction
positive
Yes. The provided context states that "U Can't Touch This" was the first single from MC Hammer's 1990 album "Please Hammer Don't Hurt 'Em" and it became a smash hit, dominating radio and MTV during 1990. It also won two Grammys that year. (✓)

negated
No. The provided context states that "U Can't Touch This" was indeed a hit record for MC Hammer in 1990. (✓)

Claude3.5 Prediction
positive
Yes. According to the text, "U Can't Touch This" was a massive hit in 1990, dominating radio and MTV. The song was part of his album "Please Hammer Don't Hurt 'Em" and won two Grammy Awards (Best RB Song and Best Solo Rap Performance). The text specifically states that the single was so popular that it would have likely topped the Billboard pop chart if not for a quirk in its release format. (✓)

negated
Yes. MC Hammer did have a hit record in 1990 with "U Can't Touch This". The passage specifically states that this song "dominated radio and MTV during 1990" and won two Grammy Awards." (✗)

Table 17: Qualitative results of vanilla method in boolean selection task. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *query negation blindness*.

Context:
Kansas City Chiefs 10 One day Lamar Hunt, architect of the AFL and owner of the Kansas City Chiefs, came across his daughter's Super Ball and was given the inspiration for the name of the championship game between the upstart American Football League and the old-guard National Football League. "Why not," he wondered, "call our championship game the Super Bowl?" The name, however, wasn't applied to this first contest until a couple years later when it was retroactively labeled. The game itself, though, caught on quickly and thus, an American tradition was born. Unlikely Hero The first of these "Super" contests pitted Vince Lombardi's Green Bay Packers (13-2) against Hank Stram's Chiefs (12-2-1) and was played before 61,946 fans in Los Angeles' 100,000-seat Memorial Coliseum. The television audience for this game is estimated to have been approximately 60 million viewers. The game itself featured an unlikely hero in Green Bay wide receiver Max McGee. McGee was strictly a backup and did not receive much playing time. In fact, in 14 games during the 1966 season he had caught only four passes for 91 yards. Legend has it that Max had spent most of the previous night out on the town and was in no shape to play football, especially in a championship game. But he felt safe in knowing the only way he would get into the game was if Boyd Dowler got hurt. "No Way!" McGee was later quoted as saying, "I waddled in about 7:30 in the morning and I could barely stand up for the kickoff. $(\cdots)$ Just moments after entering the game though, he caught a 37-yard touchdown pass from Bart Starr to cap off an 80-yard drive that gave the Packers an early lead. On the day, McGee caught seven passes for 138 yards and two touchdowns as the Packers went on to win the first Super Bowl, 35-10. Each player on the Packers received a $15,000 bonus for winning the game, while members of the Chiefs earned $7,500. A one-minute television commercial sold for $75,000 to $85,000; pocket change compared to the millions spent on air time now.

Choices:
A: Boyd Dowler
B: Kansas City Chiefs
C: Green Bay Packers
D: Memorial Coliseum

Positive Query: Which team won the first 'Super Bowl'?

Negated Query: Which team did **not** win the first 'Super Bowl'?

GT Answer: C

ChatGPT Prediction (pos/neg)
C / B (✓)

LLaMA3.1 Prediction (pos/neg)
C / C (✗)

Mistral Prediction (pos/neg)
C / C (✗)

Claude3.5 Prediction (pos/neg)
C / B (✓)

Table 18: Qualitative results of vanilla method in multiple-choice selection task. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *query negation blindness*.

Context:
The Ten Commandments (1956) on iTunes Open iTunes to preview, buy, and download this movie. About the Movie Based on the Holy Scriptures, with additional dialogue by several other hands, The Ten Commandments was the last film directed by Cecil B. DeMille. The story relates the life of Moses, from the time he was discovered in the bullrushes as an infant by the pharoah's daughter, to his long, hard struggle to free the Hebrews from their slavery at the hands of the Egyptians. Moses (Charlton Heston) starts out "in solid" as Pharoah's adopted son (and a whiz at designing pyramids, dispensing such construction-site advice as "Blood makes poor mortar"), but when he discovers his true Hebrew heritage, he attempts to make life easier for his people. Banished by his jealous half-brother Rameses (Yul Brynner), Moses returns fully bearded to Pharoah's court, warning that he's had a message from God and that the Egyptians had better free the Hebrews post-haste if they know what's good for them. Only after the Deadly Plagues have decimated Egypt does Rameses give in. As the Hebrews reach the Red Sea, they discover that Rameses has gone back on his word and plans to have them all killed. But Moses rescues his people with a little Divine legerdemain by parting the Seas. Later, Moses is again confronted by God on Mt. Sinai, who delivers unto him the Ten Commandments. Meanwhile, the Hebrews, led by the duplicitous Dathan (Edward G. Robinson), are forgetting their religion and behaving like libertines. "Where's your Moses now?" brays Dathan in the manner of a Lower East Side gangster. He soon finds out. $(\cdots)$ It is very likely the most eventful 219 minutes ever recorded to film97and who's to say that Nefertiri (Anne Baxter) didn't make speeches like, "Oh, Moses, Moses, you splendid, stubborn, adorable fool"? Credits Average Rating: 7.5/10 Top Critics' Reviews Fresh: DeMille remains conventional with the motion picture as an art form. The eyes of the onlooker are filled with spectacle. Emotional tug is sometimes lacking. – Variety Staff, Variety, Jul 7, 2010 Fresh: There is no other picture like it. There will be none. If it could be summed up in a word, the word would be sublime. And the man responsible for that, when all is said and done is Cecil B. DeMille. – James Powers, Hollywood Reporter, Apr 7, 2015 Fresh: DeMille's direction of the action is superb and the various roles are played with feeling by a large and competent cast, headed by Charlton Heston. – Kate Cameron, New York Daily News, Dec 10, 2014 Fresh: With a running time of nearly four hours, Cecil B. De Mille's last feature and most extravagant blockbuster is full of the absurdities and vulgarities one expects, but it isn't boring for a minute. – Jonathan Rosenbaum, Chicago Reader, Jun 24, 2010.

Positive Query: In the 1956 film "The Ten Commandments", Moses was played by [MASK].

Negated Query: In the 1956 film "The Ten Commandments", Moses was **not** played by [MASK].

GT Answer: Charlton Heston

ChatGPT Prediction (pos/neg)
Charlton Heston / Yul Brynner (✓)

LLaMA3.1 Prediction (pos/neg)
Charlton Heston / Charlton Heston (✗)

Mistral Prediction (pos/neg)
Charlton Heston / Yul Brynner (✓)

Claude3.5 Prediction (pos/neg)
Charlton Heston / Charlton Heston (✗)

Table 19: Qualitative results of vanilla method in cloze-style completion task. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *query negation blindness*.

Context:
Hair and skin colour | DermNet New Zealand Home » Topics A–Z » Hair and skin colour Hair and skin colour Author: Vanessa Ngan, Staff writer The differences in human skin and hair colour are among the most noticeable features of human variability. An individual's skin and hair colour provides a clue to their ancestry and heritage. Skin and hair colour is primarily determined by the genes we inherit from our parents. But what is it that makes black skin and red hair? Genetics and evolution of hair and skin colour Skin colour or pigmentation is determined by three pigments or chromophores: Melanin – a brown/black or red/yellow polymer produced by melanosomes in melanocyte cells Haemoglobin in red blood cells in the superficial vasculature Dietary carotenoids (e.g. carrots) – to a much lesser degree, and is often seen as a yellow colour on the palms Skin colour Carotenaemia Melanin content of skin is the main determining factor of skin and hair colour; hair is considered a form of skin with regards to pigmentation. Melanin is synthesized by melanosomes found in skin cells called melanocytes. Whether you have dark skin or light skin depends on the amount and type of melanin produced in your skin. There are two types of melanin and the relative amounts of each determine your skin and hair colouring. Eumelanin is responsible for producing brown or black colour Phaeomelanin is responsible for yellow or red colour. Eumelanin:phaeomelanin ratio High eumelanin and low phaeomelanin Black or dark skin High phaeomelanin and low eumelanin Light skin and freckles Red (very high phaeomelanin) or yellow None or very little eumelanin or pheomelanin (albinism) Pale White The table above gives a very simplistic explanation for skin and hair colour determination. Many other factors are involved, including a gene protein called melanocortin 1 receptor (MC1R). Increased activity of MC1R leads to the production of more eumelanin and less phaeomelanin, resulting in darkening of skin and hair. People who have impaired MC1R genes tend to have red hair and fair skin with freckles. This gene mutation increases the risk of skin cancer, particularly melanoma . Changes in gene activity associated with skin and hair colouring has been occurring since the evolution of mankind. Migration and movement of humans over the continents meant skin colour evolved quickly and readily as an adaptation to new environments. Variations of skin and hair colour Not only do we see differences in the skin and hair colour between people but sometimes we see differences in the colour between different parts of our own body. Hair colour may vary both in time and site. For example, scalp hair may be blonde in a child, then darken to brown in adolescence, and eventually become white in old age. And why in an individual can scalp hair be black or dark brown whilst facial or pubic hair be red? A number of factors are at work to explain this diversity. Ultraviolet radiation Skin exposure to sunlight (ultraviolet radiation) increases the production of melanin and can result in darker coloured skin (tanning). Repeated exposure to UVR can lead to darker skin and hair colour over time. Sun exposure can also cause brown spots and freckles , especially in fair-skinned individuals. Pigmentation disorders Pigmentation disorders can result in generalised or localised hyperpigmentation (increased skin colour) and hypopigmentation (reduced skin colour). Increase in melanin (hyperpigmentation or hypermelanosis) can be due to an increased number of pigment cells (melanocytes) or from increased production of melanin. Whilst a reduction in melanin results in pale patches (hypopigmentation or hypomelanosis) and total loss of melanin in white patches (leucoderma). Sex, age and site The number of melanocytes and the amount and type of melanin produced may be affected by body site, age and sex in the following ways: Different body sites are preprogrammed to have differing numbers of melanocytes and constitutive melanin production. Amount and type of melanin production vary with age and gender. (· · ·) With increasing age, melanocyte activity slows down and eventually stops so that melanin (pigment that gives colour to hair) is no longer produced. New hair grows without colour and is transparent hair that gives the appearance of grey hair against healthier darker hair. In some cases, grey hair may be caused by a deficiency of B12 or a thyroid imbalance. Grey hair is more obvious in people with darker hair because it stands out, but people with naturally light hair are just as likely to go grey. The genetic factors responsible for pigmentary variations needs further investigation. A better understanding of MC1R and other genes will go a long way to help in explaining the variation in skin and hair color in human populations. Related information Rees JL. Genetics of Hair and Skin Color. Annu. Rev. Genet. 2003.

Choices:
A: Haemoglobin
B: Grey hair
C: Melanin
D: Carotenoid

Positive Query: In the human body, which pigment is primarily responsible for the colour of skin?

Negated Query: In the human body, which pigment is **not** primarily responsible for the colour of skin?

GT Answer: C

LLaMA3.1-8B Prediction (pos/neg)
C / C (✗)

LLaMA3.1-70B Prediction (pos/neg)
C / C (✗)

LLaMA3.1-405B Prediction (pos/neg)
C / D (✓)

Table 20: Qualitative results on multiple-choice selection tasks across parameter size variants within LLaMa3.1 family. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *negation blindness*.

Context: Gjetost Cheese - itscheese.com cheese guide photo credit Gjetost (pronounced ýet-oast́) is a sweet whey cheese from Norway. The cheese is called Brunost in Norwegian, which means "brown cheese." The name Gjetost is used mostly in North America and is an archaic spelling of Gietost, which means "goat cheese." Gjetost has a strong, sweet taste, that is somewhat "goaty." The texture is semi-firm, often compared to the consistency of fudge. The cheese is usually formed into small rectangular blocks. Some people doní consider Gjetost to be a true cheese because of how it is produced. Gjetost is traditionally made by slowly boiling the whey of goat milk for several hours until most of the liquid is evaporated. Most Gjetost made in modern times also contains cow milk and cream in addition to goat whey. The heat turns the milk sugars into caramel, which gives the cheese itś brown color and sweet flavor. (· · ·) In Norway it is enjoyed at breakfast or as a snack food. In the United States, the two most common brands are Ski Queen and Ekte. Ski Queen is made from whey, milk, goats milk, and cream. Ekte is made from goats milk and is a little sweeter and stronger. There is also a spreadable version that is not boiled as long. Gjetost is mostly produced and consumed in Norway, but it is also made in Sweden, Iceland, and the Upper Midwest in the United States. In January 2013, a truckload of Gjetost caught fire in Norway closing a tunnel road for the 5 days it took to extinguish. (read story here) Where to buy Gjetost is slowly becoming easier to find in grocery stores in North America, especially in areas with large Scandinavian populations. We also found Gjetost online at amazon.com
Positive Query: Gjetost is the national cheese of which country?
Negated Query: Gjetost is **not** the national cheese of which country?
GT Answer: Norway
LLaMA3.1-8B Prediction (pos/neg) Norway / Norway (✗)
LLaMA3.1-70B Prediction (pos/neg) Norway / Norway (✗)
LLaMA3.1-405B Prediction (pos/neg) Norway / Iceland (✓)

Table 21: Qualitative Results on free-form generation task across size variants of LLaMa 3.1. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *query negation blindness*.

Context:
Muhammad Ali refuses to fight in Vietnam war: From the archive, 27 April 1967 | From the Guardian | The Guardian Share on Messenger Close Boxing authorities in America today stripped Muhammad Ali (Cassius Clay) of his world heavyweight title and suspended his boxing licence after he had refused to be inducted into the United States Army. Clay had stood in line with 11 other men being called up in a room in the old Post Office building in Houston, Texas, and heard his Black Muslim name called by the officer administering the oath. Clay did not move. Another officer walked up to him and said: "Mr Ali, will you accompany me, please ?" Clay did not speak, but followed him out of the room to be given a warning of the consequences of his refusal. He was taken back into the room and given a second chance to take the oath, but he again refused. $(\cdots)$ Reading on mobile? Click here to watch video At the same time a spokesman for the Justice Department said it would decide whether to ask a federal grand jury for an indictment. If an indictment were returned, Clay would have to go for trial. He could face a long prison sentence. Clay issued a statement saying: "It is in the light of my consciousness as a Muslim minister and my own personal convictions that I take my stand in rejecting the call to be inducted. I do so with the full realisation of its implications. I have searched my conscience. "I had the world heavyweight title not because it was given to me, not because of my race or religion, but because I won it in the ring. Those who want to take it and start a series of auction-type bouts not only do me a disservice, but actually disgrace themselves... Sports fans and fair-minded people throughout America would never accept such a title-holder." The New York Boxing Commission, which suspended his licence, said his refusal to enter the service was detrimental to the best interests of boxing. [Muhammad Ali was sentenced to five years in prison and a $10,000 fine, though he remained out on bail while he appealed. He was stripped of his passport and his heavyweight title and banned from fighting in the US. $(\cdots)$

Choices:
A: Black Muslim
B: The officer
C: Muhammad Ali
D: A spokesman

Positive Query: Who was stripped of his heavyweight boxing titles when he refused his US army induction in April, 1967?

Negated Query: Who was **not** stripped of his heavyweight boxing titles when he refused his US army induction in April, 1967?

GT Answer: C

Vanilla Prediction (pos/neg)
C / C (✗)

CoT Prediction (pos/neg)
C / B (✓)

Decom Prediction (pos/neg)
C / C (✗)

Refine Prediction (pos/neg)
C / A (✓)

Table 22: Qualitative results of various self reasoning-enhanced prompting methods in multiple-choice selection task using ChatGPT. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *negation blindness*.

Context:
Hair and skin colour | DermNet New Zealand Home » Topics A–Z » Hair and skin colour Hair and skin colour Author: Vanessa Ngan, Staff writer The differences in human skin and hair colour are among the most noticeable features of human variability. An individual's skin and hair colour provides a clue to their ancestry and heritage. Skin and hair colour is primarily determined by the genes we inherit from our parents. But what is it that makes black skin and red hair? Genetics and evolution of hair and skin colour Skin colour or pigmentation is determined by three pigments or chromophores: Melanin – a brown/black or red/yellow polymer produced by melanosomes in melanocyte cells Haemoglobin in red blood cells in the superficial vasculature Dietary carotenoids (e.g. carrots) – to a much lesser degree, and is often seen as a yellow colour on the palms Skin colour Carotenaemia Melanin content of skin is the main determining factor of skin and hair colour; hair is considered a form of skin with regards to pigmentation. Melanin is synthesized by melanosomes found in skin cells called melanocytes. Whether you have dark skin or light skin depends on the amount and type of melanin produced in your skin. There are two types of melanin and the relative amounts of each determine your skin and hair colouring. Eumelanin is responsible for producing brown or black colour Phaeomelanin is responsible for yellow or red colour. Eumelanin:phaeomelanin ratio High eumelanin and low phaeomelanin Black or dark skin High phaeomelanin and low eumelanin Light skin and freckles Red (very high phaeomelanin) or yellow None or very little eumelanin or pheomelanin (albinism) Pale White The table above gives a very simplistic explanation for skin and hair colour determination. Many other factors are involved, including a gene protein called melanocortin 1 receptor (MC1R). Increased activity of MC1R leads to the production of more eumelanin and less phaeomelanin, resulting in darkening of skin and hair. People who have impaired MC1R genes tend to have red hair and fair skin with freckles. This gene mutation increases the risk of skin cancer, particularly melanoma . Changes in gene activity associated with skin and hair colouring has been occurring since the evolution of mankind. Migration and movement of humans over the continents meant skin colour evolved quickly and readily as an adaptation to new environments. Variations of skin and hair colour Not only do we see differences in the skin and hair colour between people but sometimes we see differences in the colour between different parts of our own body. Hair colour may vary both in time and site. For example, scalp hair may be blonde in a child, then darken to brown in adolescence, and eventually become white in old age. And why in an individual can scalp hair be black or dark brown whilst facial or pubic hair be red? A number of factors are at work to explain this diversity. Ultraviolet radiation Skin exposure to sunlight (ultraviolet radiation) increases the production of melanin and can result in darker coloured skin (tanning). Repeated exposure to UVR can lead to darker skin and hair colour over time. Sun exposure can also cause brown spots and freckles , especially in fair-skinned individuals. Pigmentation disorders Pigmentation disorders can result in generalised or localised hyperpigmentation (increased skin colour) and hypopigmentation (reduced skin colour). Increase in melanin (hyperpigmentation or hypermelanosis) can be due to an increased number of pigment cells (melanocytes) or from increased production of melanin. Whilst a reduction in melanin results in pale patches (hypopigmentation or hypomelanosis) and total loss of melanin in white patches (leucoderma). Sex, age and site The number of melanocytes and the amount and type of melanin produced may be affected by body site, age and sex in the following ways: Different body sites are preprogrammed to have differing numbers of melanocytes and constitutive melanin production. Amount and type of melanin production vary with age and gender. $(\cdots)$ With increasing age, melanocyte activity slows down and eventually stops so that melanin (pigment that gives colour to hair) is no longer produced. New hair grows without colour and is transparent hair that gives the appearance of grey hair against healthier darker hair. In some cases, grey hair may be caused by a deficiency of B12 or a thyroid imbalance. Grey hair is more obvious in people with darker hair because it stands out, but people with naturally light hair are just as likely to go grey. The genetic factors responsible for pigmentary variations needs further investigation. A better understanding of MC1R and other genes will go a long way to help in explaining the variation in skin and hair color in human populations. Related information Rees JL. Genetics of Hair and Skin Color. Annu. Rev. Genet. 2003. 37:67–90 On DermNet NZ

Positive Query: The pigment primarily responsible for the colour of skin in the human body is [MASK].

Negated Query: The pigment primarily responsible for the colour of skin in the human body is **not** [MASK].

GT Answer: Melanin

Vanilla Prediction (pos/neg)
Melanin / Melanin (✗)

CoT Prediction (pos/neg)
Melanin / Haemoglobin (✓)

Decom Prediction (pos/neg)
Melanin / Melanin (✗)

Refine Prediction (pos/neg)
Melanin / Melanin (✗)

Table 23: Qualitative results of various mitigation prompting methods in cloze-style completion task using ChatGPT. ✓ indicates that the model has maintained logical consistency and correctly recognized negation. ✗ indicates the occurrence of *query negation blindness*.

D Prompt Templates

Table 24-27 show the prompt templates used in the data construction processes for the designed tasks (§ 3.4).

We verify the *negation blindness* on the query of LLMs through four verification tasks, i.e., boolean selection, multiple-choice selection, cloze-style completion, and free-form generation. Table 28-31 show the vanilla prompt templates used in the validation experiments (§ 5.1) for each designed task.

Prompt templates for evaluating the results of generating LLMs are Table 32 and 33, which indicate text entailment evaluation and LLM-as-a-Judge prompts, respectively.

Moreover, we explore the mitigation feasibility of the *negation blindness* problem (§ 5.2) to provide extended insights and understanding, and the prompt templates used for this are Table 34-46. Specifically, Table 34-37, Table 38-41, Table 42-45, Table 46 represent chain-of-thought, task decomposition, self-refine, and multi-agents based method prompt templates, respectively.

Task Instruction

```
You are a helpful assistant that transforms a question and its corresponding gold answer into a declarative sentence. The resulting sentence should convey the same meaning as the original question-answer pair in a clear and natural way.

Input Format:
# Question: [Insert question here]
# Gold Answer: [Insert gold answer here]

Output Format:
# Declarative Sentence: [Your response here]

Example:
# Question: What is the capital of France?
# Gold Answer: Paris

# Declarative Sentence: The capital of France is Paris.
```

```
Input: Now, please process them according to the following format:
# Question: {{question}}
# Gold Answer: {{answer}}

# Declarative Sentence:
```

Table 24: Prompt template for the transformation into proposition sentence.

Task Instruction

```
Given a positive question, the objective is to transform it into its negative counterpart.
Specifically, this transformation requires semantic negation of the gold answer phrase (or entity) ‘X’.
To achieve this, an appropriate negation cue ‘no’ must be inserted directly before ‘X’.
Importantly, the transformation should preserve the original sentence structure as much as possible.
The resulting negative question must explicitly reflect the negation by replacing the gold answer ‘X’ with ‘no X’.

Input Format:
# Positive question: [Insert sentence here]
# Gold label: [Insert ground-truth answer here]

Output Format:
# Negative question: [Your response here]

Examples:
# Positive question: Did Rita Coolidge sing the title song for the Bond film Octopussy?
# Gold Answer: Octopussy
# Negative interrogative sentence: Did Rita Coolidge sing the title song for the Bond film {no Octopussy}?

# Positive Question: Was Kiefer Sutherlund born before Christian Slater?
# Gold Answer: Kiefer Sutherlund
# Negative Question: Was {no Kiefer sutherlund} was never born before Christian Slater.
```

```
Input: Now, please process it according to the following format:
# Positive question: {{input_sentence}}

# Negative question:
```

Table 25: Prompt template for boolean selection task construction.

Task Instruction

Transform the given affirmative sentence into a negative sentence. Maintain the original sentence structure as much as possible and ensure grammatical accuracy by correctly inserting 'not'.

Input Format:
Affirmative Sentence: [Insert question here]

Output Format:
Negative Sentence: [Your response here]

Example:
Affirmative Sentence: [MASK] was born before Christian Slater.

Negative Sentence: [MASK] was not born before Christian Slater.

Input:
Now, please process the following input:
Affirmative Sentence: {{input_sentence}}

Negative Sentence:

Table 26: Prompt template for cloze-style completion task construction.

Task Instruction

Transform the given interrogative sentence into a negative interrogative sentence. Maintain the original sentence structure as much as possible and ensure grammatical accuracy by correctly inserting 'not'.

Input Format:
Interrogative sentence: [Insert sentence here]

Output Format:
Negative interrogative sentence: [Your response here]

Examples:
Interrogative sentence: Which Scottish newspaper features the Broons and Oor Wullie?
Negative interrogative sentence: Which Scottish newspaper does not feature the Broons and Oor Wullie?

Interrogative sentence: What was Blondie's last UK No 1 of the 80s?
Negative interrogative sentence: What was not Blondie's last UK No 1 of the 80s?

Input:
Now, please process the following input:
Interrogative sentence: {{input_sentence}}

Negative interrogative sentence:

Table 27: Prompt template for free-form generation task construction.

Task Instruction

You aim to generate an appropriate answer to the given question.
'{not X}' in the question indicates 'someone(thing) other than X'.
You should respond using "Yes" or "No".
Also, please provide a brief rationale for your response.

{{exemplar}}

Please respond according to the given format:
—
Context: {{context}}
Question: {{question}}

Your Answer:
Rationale:

Table 28: Vanilla prompt template for boolean selection task.

Task Instruction

You aim to choose an appropriate option to the given question.
You should respond using "A", "B", "C", or "D" because this is a multiple choice question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format:
—
Context: {{context}}
Question: {{question}}
Choices: {{choices}}

Your Answer:

Table 29: Vanilla prompt template for multiple-choice selection task.

Task Instruction

```
 You aim to generate an answer to the given
question.
You should only generate words or phrases that
fit the [MASK] part.
Only generate words or phrases for [MASK].
Please do not repeat the given question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given
format:
—
# Context: {{context}}
# Question: {{question}}

# Your Answer ([MASK]):
```

Table 30: Vanilla prompt template for cloze-style completion task.

Task Instruction

```
 You aim to generate an answer to the given
question.
You should respond using short-form answer
format.
Please do not generate any other descriptions.
Please generate a short answer only.

{{exemplar}}

Please respond according to the given
format:
—
# Context: {{context}}
# Question: {{question}}

# Your Answer:
```

Table 31: Vanilla prompt template for free-form generation task.

Task Instruction

```
 Your goal is to determine the similarity
or agreement between the responses of two
assistants.
You aim to perform a task similar to natural
language inference (NLI).
Determine whether the two given responses are
semantically related to each other in terms
of “entailment” or “contradiction.”
All options in the Assistant B’s answer list
can be treated as aliases.
Output your final classification by strictly
following this format: "[[entailment]]" if
the two responses are semantically similar
or identical, "[[contradiction]]" When two
responses are semantically opposite or
contradictory.
Please do not generate any other descriptions.
—
# User Question:
{{question}}

# The Start of Assistant A’s Answer
{{answer_a}}
# The End of Assistant A’s Answer

# The Start of Assistant B’s Answer
{{answer_b}}
# The End of Assistant B’s Answer

# Your Evaluation:
```

Table 32: Text entailment evaluation prompt template for the LLMs’ generated responses.

Task Instruction

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.
You should choose the assistant that responds with better consideration to the user's question and the preferred answer.
Your evaluation should consider factors such as the helpfulness, relevance, and accuracy of their responses.
Begin your evaluation by comparing the two responses and provide a short explanation.
Do not favor certain names of the assistants.
Be as objective as possible.
After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.
—
# User Question:
{{question}}

{{choices}}

# Preferred Answer:
{{gold}}

# The Start of Assistant A's Answer
{{answer_a}}
# The End of Assistant A's Answer

# The Start of Assistant B's Answer
{{answer_b}}
# The End of Assistant B's Answer


# Your Evaluation:
```

Table 33: Prompt template for LLM-as-a-Judge method using pair-wise comparison.

Task Instruction

```
Step 1
You aim to generate an appropriate answer to the given question.
'{not X}' in the question indicates something other than X.
You should respond using "Yes" or "No".
Also, please provide a brief rationale for your response.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:
# Rationale:
##
First, you should generate reasoning paths before generating a final response. "Let's think step-by-step."

Step 2
# Reasoning paths you made:
{{reasoning_path}}

Respond appropriately referring to the given reasoning paths.
```

Table 34: Chain-of-thought (CoT) prompt template for boolean selection task.

Task Instruction

```
Step 1
You aim to choose an appropriate option to the given question.
You should respond using "A", "B", "C", or "D" because this is a multiple choice question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Choices: {{choices}}
# Your Answer:
##
First, you should generate reasoning paths before generating a final response. "Let's think step-by-step."

Step 2
# Reasoning paths you made:
{{reasoning_path}}

Respond appropriately referring to the given reasoning paths.
Please do not generate any other descriptions.
```

Table 35: Chain-of-thought (CoT) prompt template for multiple-choice selection task.

Task Instruction

```
Step 1
You aim to generate an answer to the given question.
You should only generate words or phrases that fit the [MASK] part.
Only generate words or phrases for [MASK].
Please do not repeat the given question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer ([MASK]):
##
First, you should generate reasoning paths before generating a final response. "Let's think step-by-step."

Step 2
# Reasoning paths you made:
{{reasoning_path}}

Respond appropriately referring to the given reasoning paths.
Please do not generate any other descriptions.
```

Table 36: Chain-of-thought (CoT) prompt template for cloze-style completion task.

Task Instruction

```
Step 1
You aim to generate an answer to the given question.
You should respond using short-form answer format.
Please do not generate any other descriptions.
Please generate a short answer only.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:
##
First, you should generate reasoning paths before generating a final response. "Let's think step-by-step."

Step 2
# Reasoning paths you made:
{{reasoning_path}}

Respond appropriately referring to the given reasoning paths.
Please do not generate any other descriptions.
Please generate a short answer only.
```

Table 37: Chain-of-thought (CoT) prompt template for free-form generation task.

Task Instruction

```
Step 1
You aim to generate an appropriate answer to the given question.
'{not X}' in the question indicates something other than X.
You should respond using "Yes" or "No".
Also, please provide a brief rationale for your response.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:
# Rationale:
##
First, you should decompose the given claim into sub-claims before generating a final response. "Let's break down the claim into sub-claims."

Step 2
# Sub-claims you made:
{{reasoning_path}}

Respond appropriately referring to the given sub-claims.
```

Table 38: Decomposition prompt template for boolean selection task.

Task Instruction

```
Step 1
You aim to choose an appropriate option to the given question.
You should respond using "A", "B", "C", or "D" because this is a multiple choice question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Choices: {{choices}}
# Your Answer:
##
First, you should decompose the given claim into sub-claims before generating a final response. "Let's break down the claim into sub-claims."

Step 2
# Sub-claims you made:
{{reasoning_path}}

Respond appropriately referring to the given sub-claims.
Please do not generate any other descriptions.
```

Table 39: Decomposition prompt template for multiple-choice selection task.

Task Instruction

Step 1
You aim to generate an answer to the given question.
You should only generate words or phrases that fit the [MASK] part.
Only generate words or phrases for [MASK].
Please do not repeat the given question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format.
—
Context: {{context}}
Question: {{question}}
Your Answer:
##
First, you should decompose the given claim into sub-claims before generating a final response. "Let's break down the claim into sub-claims."

Step 2
Sub-claims you made:
{{reasoning_path}}

Respond appropriately referring to the given sub-claims.
Please do not generate any other descriptions.

Table 40: Decomposition prompt template for cloze-style completion task.

Task Instruction

Step 1
You aim to generate an answer to the given question.
You should respond using short-form answer format.
Please do not generate any other descriptions.
Please generate a short answer only.

{{exemplar}}

Please respond according to the given format.
—
Context: {{context}}
Question: {{question}}
Your Answer:
##
First, you should decompose the given claim into sub-claims before generating a final response. "Let's break down the claim into sub-claims."

Step 2
Sub-claims you made:
{{reasoning_path}}

Respond appropriately referring to the given sub-claims.
Please do not generate any other descriptions.
Please generate a short answer only.

Table 41: Decomposition prompt template for free-form generation task.

Task Instruction

```
Step 1
You aim to generate an appropriate answer to
the given question.
'not X' in the question indicates something
other than X.
You should respond using "Yes" or "No".
Also, please provide a brief rationale for
your response.

{{exemplar}}

Please respond according to the given
format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:
# Rationale:

Step 2
# Previous response you made:
{{reasoning_path}}

You should provide "feedback" on whether
your previous response was appropriate and
whether any part needs to be refined.

# Feedback:

Step 3
# Previous response you made:
{{reasoning_path}}
# Feedback on the previous response:
{{feedback}}
Respond appropriately referring to the
responses you have previously made and the
feedback on them.
```

Table 42: Refine prompt template for boolean selection task.

Task Instruction

```
Step 1
You aim to choose an appropriate option to the
given question.
You should respond using "A", "B", "C", or "D"
because this is a multiple choice question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given
format.
—
# Context: {{context}}
# Question: {{question}}
# Choices: {{choices}}
# Your Answer:

Step 2
# Previous response you made:
{{reasoning_path}}

You should provide "feedback" on whether
your previous response was appropriate and
whether any part needs to be refined.

# Feedback:

Step 3
# Previous response you made:
{{reasoning_path}}
# Feedback on the previous response:
{{feedback}}
Respond appropriately referring to the
responses you have previously made and the
feedback on them.
Please do not generate any other descriptions.
```

Table 43: Refine prompt template for multiple-choice selection task.

Task Instruction

```
Step 1
You aim to generate an answer to the given question.
You should only generate words or phrases that fit the [MASK] part.
Only generate words or phrases for [MASK].
Please do not repeat the given question.
Please do not generate any other descriptions.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:

Step 2
# Previous response you made:
{{reasoning_path}}

You should provide "feedback" on whether your previous response was appropriate and whether any part needs to be refined.

# Feedback:

Step 3
# Previous response you made:
{{reasoning_path}}
# Feedback on the previous response:
{{feedback}}
Respond appropriately referring to the responses you have previously made and the feedback on them.
Please do not generate any other descriptions.
```

Table 44: Refine prompt template for cloze-style completion task.

Task Instruction

```
Step 1
You aim to generate an answer to the given question.
You should respond using short-form answer format.
Please do not generate any other descriptions.
Please generate a short answer only.

{{exemplar}}

Please respond according to the given format.
—
# Context: {{context}}
# Question: {{question}}
# Your Answer:

Step 2
# Previous response you made:
{{reasoning_path}}

You should provide "feedback" on whether your previous response was appropriate and whether any part needs to be refined.

# Feedback:

Step 3
# Previous response you made:
{{reasoning_path}}
# Feedback on the previous response:
{{feedback}}
Respond appropriately referring to the responses you have previously made and the feedback on them.
Please do not generate any other descriptions.
Please generate a short answer only.
```

Table 45: Refine prompt template for free-form generation task.

Task Instruction
Debate You will debate your argument based on the opinions of Assistants A, B, and C regarding the given context and question. Briefly present your arguments for A, B, and C, including whether you agree or disagree. Please answer briefly. {{exemplar}} Please respond according to the given format. — # Context: {{context}} # Question: {{question}} # Opinions: {{opinions}} # Your Argument: Moderator You goal is to generate an appropriate answer to the given context and question, referring to the several assistants's (A, B, and C) opinions and arguments into account. After providing your explanation, output your final verdict by strictly following this format: [[Your Answer]]

Table 46: Prompt template example for multi-agents debate method.