Likelihood as a Performance Gauge for Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Recent work finds that retrieval-augmented generation with large language models is prone to be influenced by the order of retrieved documents in the context. However, the lack of in-depth analysis limits the use of this phenomenon for prompt engineering in practice. In this study, we posit that likelihoods serve as an effective gauge for language model performance. Through experiments on two question-answering datasets with a variety of state-of-the-art language models, we reveal correlations between answer accuracy and the likelihood of the question at both the corpus level and the instance level. In addition, we find that question likelihood can also indicate the position of the task-relevant information in the context. Based on these findings, we propose two methods that use question likelihood as a gauge for selecting and constructing prompts that lead to better performance. Experiments show that our likelihood-based methods are effective and also efficient, as they only need to compute the likelihood of the input, requiring much fewer language model passes than heuristic prompt engineering methods that require generating responses. Our analysis deepens our understanding of how input prompts affect model performance and provides a promising direction for efficient prompt optimization.[1]

1 Introduction

Prompt designing is crucial for large language models (LMs) when tackling downstream tasks with retrieval-augmented generation (RAG, Lewis et al., 2020). Well-designed prompts can boost LMs' performance and lead them to generate responses that better meet users' expectations (Gao et al., 2021; Izacard et al., 2024; Liu et al., 2024; Schulhoff et al., 2024; Ma et al., 2024, *inter alia*).

Typically, under the RAG framework, a prompt consists of three major components—an **instruction** defining a task and providing general guidance, a specific **question**[2] of the task, and a **context** comprising a set of documents retrieved by retrievers from some external source (Karpukhin et al., 2020; Ni et al., 2022). Much previous work has explored empirical approaches of prompt engineering, such as manually designing prompts that mimic human reasoning (Wei et al., 2023; Yao et al., 2023). Recently, Liu et al. (2024) have shown that LM performance is substantially affected by the order of the retrieved documents in the context: Namely, the answer accuracy peaks when the gold document[3] is placed at the beginning or the end of the context. While extensive experimental results

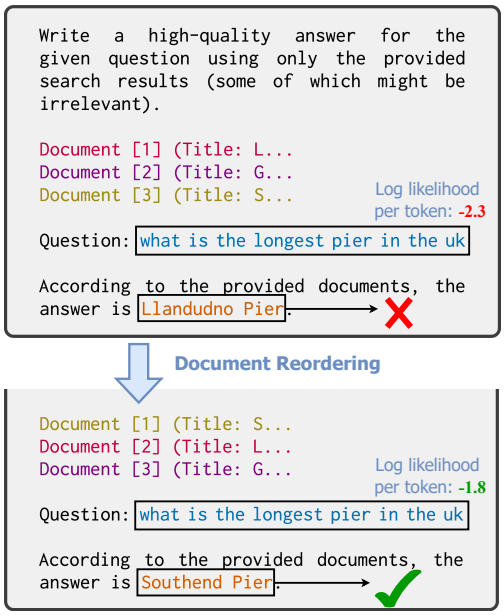


Figure 1: A prompt with higher question likelihood tends to lead to a better answer.

[1]Our code is available at `https://anonymous`.

[2]Formally, in an input prompt, we refer to the segment that directly conveys the question or query expected to be solved by the LM's output as a question. In our experiments, this contains the entire question sentence, including the punctuation.

[3]In factual QA tasks, the document containing the ground truth answer as a substring is referred to as a "gold document".

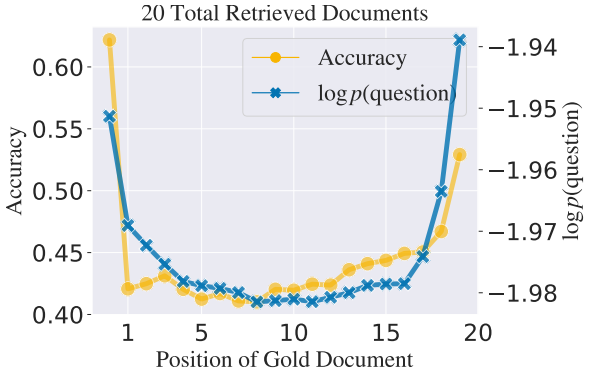


Figure 2: Besides answer accuracy (Liu et al., 2024), we find that the likelihood of question *simultaneously* fluctuates in a U-shaped curve as the gold document position within the context changes. The log-likelihood is computed per token in the question, with `LLaMA-3-8B`.

have been presented for validating the existence of such a phenomenon, there is a lack of insights into the underlying mechanism driving it, limiting its applicability in prompt designing and optimization for real-world applications.

In this work, we contend that the likelihood assigned by the LM to a question preceded by a given retrieval-augmented context can provide useful information on forecasting the LM's accuracy in answering that question. For verification, we conduct comprehensive experiments on two QA benchmarks (NQ-Open and ELI5) with a variety of state-of-the-art open LMs (`LLaMA-2`, `LLaMA-3`, `LLaMA-3.1`, `Mistral-v0.3`, `MPT`). We focus on three functional components in the input prompt and LM output—namely the context, the question, and the gold answer—and analyze their log-likelihood.[4] At the corpus level, we find that LMs can better respond to questions with higher log-likelihoods; at the instance level, contexts with favorable document orders that lead to higher likelihoods of a particular question are also more likely to elicit better answers as shown in Figure 1.

Besides, we find that changing the position of relevant information in the input context *simultaneously* affects question likelihood and answer accuracy, as illustrated in Figure 2. Thus, question likelihood can be deemed both a *performance gauge* and a strong indicator of the position of useful task-relevant information in the input context. Based on these findings, we propose a promising direction for prompt optimization with two specific methods. The first directly takes the prompt that leads to the highest question likelihood during random document shuffling. The other uses question likelihood as a gauge for task-relevant information and reorders the documents within the context to get a better prompt. Experiments show that our methods improve answer accuracy on the two datasets for both instruction-tuned and base models. In addition to effectiveness, our method is efficient because it *only* employs the encoding function of pretrained LMs and computes the likelihood for each token in the prompt. Because LM encoding can be parallelized, the computation time for encoding can be vastly shorter than decoding an LM response.[5]

To our knowledge, our work is the first to present in-depth analyses of the relation between question likelihood and model performance under the RAG framework. Our contributions in this paper are summarized as follows:

- We hypothesize and prove that question likelihood positively correlates with answer accuracy at corpus level on NQ-Open and ELI5.
- We also demonstrate a strong instance-level correlation and verify its generality on the two datasets, based on our hypothesis.
- We reveal that question likelihood is an indicator of the position of task-relevant information in the context.
- We validate the effectiveness and efficiency of using question likelihood as a gauge for prompt optimization and demonstrate that likelihood-based prompt optimization is a promising direction for future study.

2 Related Work

2.1 Prompt Engineering

Prompt engineering is important for making the best use of LMs in real-world applications (Giray, 2023; Ekin, 2023; Gonen et al., 2023). The most straightforward prompt engineering method is to manually design prompts using heuristics, which requires human experts to design prompts based on domain-specific knowledge and select the prompts that lead to better performance on downstream tasks (Zhou et al., 2023; Marvin et al., 2023). Meanwhile, another line of work explores automatic approaches for prompts engineering

[4] In information theory, the negative log-likelihood is also called **surprisal** as it quantifies how "surprising" a particular outcome is to the estimator.

[5] Encoding, also known as **prefill phase** or **prompt phase**, requires significantly fewer LM passes than decoding. Previous work (Kwon et al., 2023; Zhong et al., 2024) reports that the throughput of processing input prompts, measured by the number of tokens processed per second, can be up to three orders of magnitude larger than that of completion generation.

(Gao et al., 2021; Pryzant et al., 2023). However, they both require decoding for outputs from LMs to evaluate the quality of prompts, thus incurring high computational costs.

2.2 Retrieval-Augmented Generation

Retrieval-augmented generation is a promising technique for improving LMs' ability to solve knowledge-intensive tasks (Lewis et al., 2020; Asai et al., 2021; Borgeaud et al., 2022). In the RAG framework, a set of documents relevant to a user query is retrieved from an external source and inserted into prompts as a context, to provide additional information to the LM and improve response quality (Petroni et al., 2020; Lewis et al., 2020). RAG tasks can be divided into two types: short-form and long-form, depending on the topic of the questions and the format of the expected answers. Short-form QA (Izacard and Grave, 2021; Liu et al., 2024) usually concerns factual questions about real-world facts. The expected answers are often unambiguous and concrete words or short phrases. Long-form QA (Fan et al., 2019; Gao et al., 2023) involves *how*, *why*, and *what* questions that seek more comprehensive responses.

2.3 Effect of Document Order on Answer Accuracy

Liu et al. (2024) finds that LMs perform better when the document with relevant information is positioned at the beginning or the end of the prompt using under RAG framework.[6] Specifically, when moving the task-relevant information from the beginning to the end of the document sequence, answer accuracy exhibits a U-shaped trend on a multi-document QA task and a synthetic key–value retrieval task, both using RAG pipelines. However, Liu et al. (2024) mainly focuses on an empirical study with less in-depth analysis, resulting in a gap between the phenomenon and its practical implications. In this work, we attempt to bridge this gap.

3 Experimental Setup

3.1 Datasets

NQ-Open. We first experiment on the NQ-Open dataset following Liu et al. (2024). This dataset covers 2655 factual questions curated from the Natural Questions dataset (Kwiatkowski et al., 2019; Lee et al., 2019) under `CC-BY-SA-3.0` license. Each question is accompanied by k documents retrieved from Wikipedia, among which *exactly one* contains the answer to the question, namely the gold document. The remaining $k-1$ documents are termed **distractors**, which are relevant to the topic of the question but do not contain any ground truth answers, retrieved using Contriever (Izacard et al., 2022). In our experiments, the total number of documents k is taken to be $\{10, 20, 30\}$.[7]

ELI5. To validate the generality of our findings, we also experiment on an open-ended non-factual QA dataset ELI5 (Fan et al., 2019) with `BSD` license. ELI5 consists of questions beginning with *how*, *why* or *what* curated from the Reddit forum "Explain Like I'm Five"[8], where the answers are expected to be more comprehensive and diverse. Each question is accompanied by k documents retrieved from Sphere (Piktus et al., 2021)—a filtered version of Common Crawl[9], where k is taken to be $\{5, 10, 20\}$ to avoid truncation due to the long questions and LMs responses for the long-form QA task. In contrast to NQ-Open, ELI5 does not provide the annotations of gold documents, which aligns with real-world RAG application scenarios, making it a more practical and challenging dataset (Nakano et al., 2021; Menick et al., 2022; Liu et al., 2023).

3.2 Answer Accuracy Metrics

NQ-Open and ELI5 apply different evaluation metrics to the LM responses. Examples are illustrated in Appendix A.

On NQ-Open, the ground truth answer for each question is either a word or a short phrase. The accuracy is 1 when the LM response contains the golden answer as a sub-string, otherwise the accuracy is 0. Following Liu et al. (2024), we compute the model's average accuracy over the entire dataset. On ELI5, the ground truth answer for each question comprises three sub-claims. An ideal LM response is expected to entail all of these claims. We follow Gao et al. (2023) and take $r/3$ to be the accuracy for each instance, where r is the number of recalled sub-claims entailed by LM responses. TRUE model[10], a `T5-XXL` model

[6] In Liu et al.'s (2024) experimental settings, the gold document is unique in a prompt for each question.

[7] We remark that NQ-Open was specifically synthesized to examine how answer accuracy is affected by changing the position of relevant information. In real-world applications, the retrieved documents for one question may contain multiple gold documents or none. Nevertheless, it mimics the RAG setup underlying many commercial generative search and QA systems.

[8] `https://www.reddit.com/r/explainlikeimfive/`

[9] `https://commoncrawl.org`

[10] `https://huggingface.co/google/t5_xxl_true_nl`

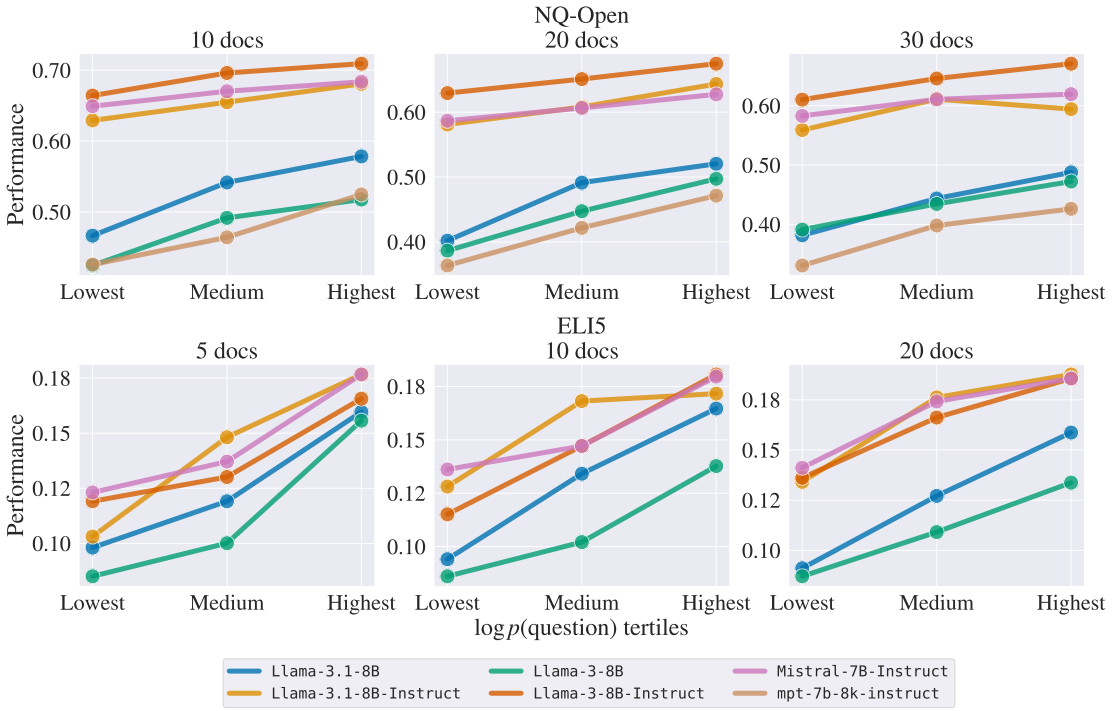


Figure 3: Corpus-level correlation between $\log p(\text{question})$ and answer accuracy on NQ-Open and ELI5.

fine-tuned on natural language inference (NLI) tasks, is used to automatically evaluate whether a response entails a sub-claim.

3.3 Language Model Settings

Models. Most state-of-the-art closed LMs, e.g., OpenAI's `ChatGPT`, `GPT-4` and Anthropic's `Claude`, do not provide direct access to the likelihood of either input or output tokens. Thus, we select the state-of-the-art open LMs for our experiments, including three families of open LMs, namely `LLaMA-2`, `LLaMA-3` (Touvron et al., 2023), and `Mistral-v0.3` (Jiang et al., 2023). Besides, we also evaluate `MPT` on NQ-Open.[11] Following the settings of Liu et al. (2024), we adopt greedy decoding for all models when generating responses. The maximum number of decoded tokens is set to 100 on NQ-Open and 300 on ELI5.

Prompt Templates. We follow the suggested usage and prompt formatting instructions of each LM we use. For chat and instruction-finetuned models, we present the context and query to the LM in the role of `user`, and elicit the response from LMs in the role of `assistant`. For base models, we elicit responses from LMs as sentence completion.

`i_mixture`

[11] In our preliminary experiments, `MPT` fails to generate adequately long responses on ELI5, resulting in an incomparable performance to other LMs.

4 Question Likelihood Correlates with Model Performance

As introduced in Section 1, we look into question likelihood as the position of the gold document changes. Intuitively, it is reasonable to expect a strong correlation between question likelihood and answer likelihood. I.e., a high question likelihood often signifies the answer to it having a high likelihood as they tend to co-occur in the same sentence during training. Meanwhile, answer likelihood is expected to correlate with task performance. Thereby, question likelihood correlates with task performance.

4.1 Likelihood Measurement

In the prompt templates used in our experiments, question is located after the retrieved documents, i.e., the context, in the input prompt (see the example in Appendix B). Thus, its likelihood can be affected by changes in preceding documents due to the autoregressive nature of LMs. To this end, we look into question likelihood and calculate the question likelihood per token

$$\log p(\text{question})_{\text{avg}} = \frac{\sum_{n=q_{\text{start}}}^{q_{\text{end}}} \log p_{\text{LM}}(w_n)}{q_{\text{end}} - q_{\text{start}} + 1},$$

where q_{start} and q_{end} represent the indices of the first and the last token in the tokenized question.

4.2 Corpus-Level Correlation

Following previous works (Gao et al., 2023; Liu et al., 2024), the answer accuracy metrics we use in our tasks (cf. Section 3.2) produce fixed discrete values from $\{0, 1\}$ or $\{0, 0.33, 0.67, 1\}$ on NQ-Open and ELI5, respectively. Thus, commonly-used metrics such as Pearson's and Spearman's correlation coefficients (Pearson, 1895; Spearman, 1904) between answer accuracy and question likelihood of these instances may not be adequately informative. Therefore, for each LM, we group the instances by the three tertiles (denoted by lowest, medium, highest) based on their $\log p(\text{question})$ and compute the average answer accuracy for each group. Results in Figure 3 show that LMs tend to perform better on the prompts with a higher $\log p(\text{question})$ compared to those with lower $\log p(\text{question})$, demonstrating that the likelihood of question is indicative of answer accuracy at the corpus level.

4.3 Instance-Level Correlation

We further analyze the instance-level correlation between $\log p(\text{question})$ and answer accuracy by varying context while keeping the question fixed.

4.3.1 Revisiting Positional Bias on NQ-Open

To begin with, we experiment on NQ-Open by varying the position of the gold document[12] in the context. The set of retrieved documents and the order of non-gold documents remain the same. As the gold document moves across different positions within the context, both $\log p(\text{question})$ and answer accuracy fluctuate. Using the same experimental setup, Liu et al. (2024) observed a drop in answer accuracy when the gold document is positioned within the middle of the context, an LM positional bias termed "lost in the middle". We replicate the similar U-shaped curve of answer accuracy in Figure 4.

To study the correlation between likelihood and answer accuracy, we plot the trend of the question likelihoods in the same Figure 4, normalized by

$$n_i = \frac{\ell_i - \min_{j \in \{1,2,\cdots,k\}} \ell_j}{\max_{j \in \{1,2,\cdots,k\}} \ell_j - \min_{j \in \{1,2,\cdots,k\}} \ell_j},$$

where $\{\ell_1, \cdots, \ell_k\}$ denotes the $\log p(\text{question})$ averaged over the dataset when the gold document is placed at position $\{1, \cdots, k\}$. The normalized value n_i falls in the range $[0, 1]$ for all $i \in \{1, 2, \cdots, k\}$, enabling better visualization and facilitating comparison among different setups. Also, we compute the trend of the normalized log-likelihood of the ground truth answers $\log p(\text{answer})$, which directly reflects the model's ability to generate the correct answer. For each question, we append the gold answer to the prompt and compute the $\log p(\text{answer})$. [13]

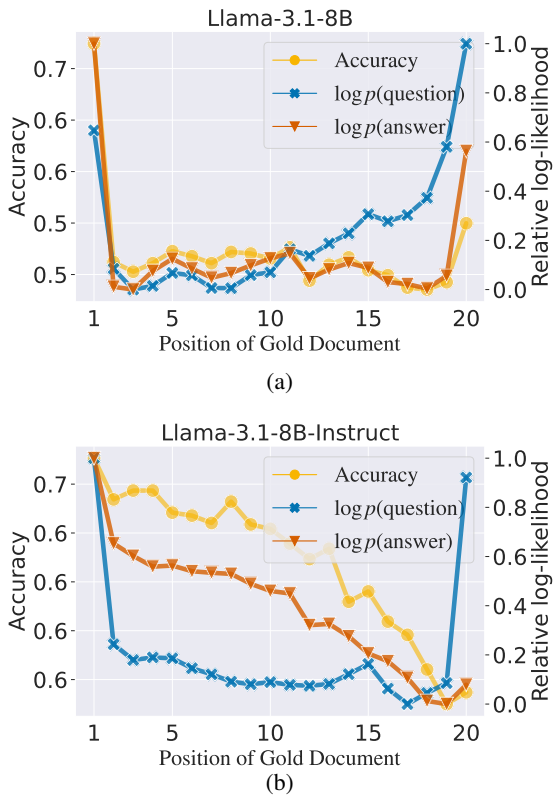


Figure 4: Relative question and answer log-likelihoods on 20 docs evaluated on `LLaMA-3.1-8B` and `LLaMA-3.1-8B-Instruct`.

Illustrated in Figure 4, the simultaneous changes in $\log p(\text{question})$ and answer accuracy strongly support our claim of the instance-level correlation. Besides, a U-shaped curve of $\log p(\text{answer})$ is observed, aligning with both $\log p(\text{question})$ and answer accuracy.[14] These observations provide a *potential explanation* of the instance-level correlation: For each instance, a higher question likelihood indicates that the input prompt is less surprising to LMs, implying that they are likely to have seen similar prompts during pretraining, with a gold document positioned at the beginning or end

[12] Recall that exactly one retrieved document is marked as the gold document for each question in NQ-Open.

[13] For instance, we append the sentence "`According to the provided documents, the answer is Southend Pier.`" directly after the question "`what is the longest pier in the uk`".

[14] See Appendix C for a complete illustration of the correlation under $\{10, 20, 30\}$-document setups.

#Doc	$\log p$(question)	Mistral-7B-Inst	LLaMA-3-8B	LLaMA-3.1-8B	LLaMA-3-8B-Inst	LLaMA-3.1-8B-Inst	MPT-7B-8K-Inst
NQ-Open							
10	Highest	**68.69** (-2.52)	**54.04** (-1.84)	**56.72** (-2.41)	**71.58** (-1.80)	**66.13** (-2.16)	**48.93** (-2.80)
	Lowest	66.98 (-2.89)	49.30 (-2.03)	53.29 (-2.72)	71.29 (-2.01)	65.70 (-2.43)	46.97 (-3.38)
20	Highest	**64.86** (-2.45)	**52.05** (-1.99)	**52.50** (-2.40)	**69.00** (-1.83)	**62.97** (-2.21)	**42.25** (-2.70)
	Lowest	62.60 (-2.83)	46.91 (-2.03)	48.51 (-2.72)	67.68 (-2.01)	61.05 (-2.43)	42.09 (-3.23)
30	Highest	**57.70** (-2.52)	**50.30** (-1.88)	**50.00** (-2.60)	64.36 (-1.84)	**60.95** (-2.41)	**39.31** (-2.56)
	Lowest	53.96 (-2.92)	45.27 (-2.03)	46.42 (-2.83)	**65.12** (-2.03)	59.55 (-2.65)	39.12 (-3.05)

Table 1: Instance-level correlation between $\log p$(question) and answer accuracy. We compute the average answer accuracy over prompts that yield the highest and lowest $\log p$(question) as the gold document placed at different positions in the document sequence for each instance. The answer accuracy and the average $\log p$(question) are reported in the table.

#Doc	$\log p$(question)	Mistral-7B-Inst	LLaMA-3-8B	LLaMA-3.1-8B	LLaMA-3-8B-Inst	LLaMA-3.1-8B-Inst
ELI5 with Rotational Reordering						
5	Highest	**13.97** (-3.72)	**11.37** (-2.23)	**12.60** (-2.28)	**14.23** (-2.21)	**13.97** (-2.26)
	Lowest	13.50 (-4.06)	11.10 (-2.39)	12.50 (-2.43)	13.17 (-2.54)	13.93 (-2.48)
10	Highest	**15.23** (-3.53)	11.27 (-2.19)	12.50 (-2.29)	**14.50** (-2.10)	**16.17** (-2.23)
	Lowest	14.47 (-3.99)	**11.50** (-2.39)	**13.10** (-2.48)	14.07 (-2.55)	15.77 (-2.54)
20	Highest	**16.20** (-2.13)	11.13 (-2.19)	**12.77** (-2.28)	**16.20** (-2.13)	**17.17** (-2.18)
	Lowest	15.80 (-2.73)	**11.20** (-2.42)	12.13 (-2.48)	15.80 (-2.73)	15.67 (-2.54)
ELI5 with Random Shuffling						
5	Highest	**14.27** (-3.73)	10.73 (-2.24)	**12.57** (-2.28)	**14.10** (-2.23)	**14.20** (-2.27)
	Lowest	14.10 (-4.04)	**11.20** (-2.39)	12.33 (-2.42)	12.77 (-2.52)	14.00 (-2.48)
10	Highest	**15.63** (-3.54)	**11.47** (-2.19)	**12.73** (-2.29)	**15.70** (-2.11)	**16.90** (-2.23)
	Lowest	15.07 (-3.97)	11.23 (-2.39)	12.20 (-2.48)	14.57 (-2.52)	16.70 (-2.53)
20	Highest	16.10 (-3.44)	10.83 (-2.19)	**12.60** (-2.28)	**16.13**(-2.14)	**17.20** (-2.18)
	Lowest	**16.53** (-4.00)	**11.20** (-2.42)	11.87 (-2.49)	15.53 (-2.71)	17.10 (-2.54)

Table 2: Instance-level correlation between $\log p$(question) and answer accuracy on ELI5. The average answer accuracy is computed over prompts that yield the highest and lowest $\log p$(question) as the input documents are reordered with (1) rotational reordering and (2) random shuffling as introduced in Section 4.3.2. The answer accuracy and the average $\log p$(question) are reported in the table.

of the document sequence.[15] Thus, the model tends to assign a higher probability to the gold answer, leading to better answer accuracy after decoding.

To further verify our finding, we calculate the average answer accuracy with the prompt of the highest and lowest question likelihoods. Results in Table 1 also show a strong instance-level correlation between $\log p$(question) and answer accuracy across almost all models and the number of documents in the question. Models are more likely to perform better when the document order in the prompt leads to the highest $\log p$(question); while the prompt with the lowest $\log p$(question) results in inferior performance in most cases.

4.3.2 Experiments on ELI5

To validate the generality of the instance-level correlation, we experimented with ELI5. Compared with NQ-Open, ELI5 is a more challenging long-form QA dataset where questions are mostly about *how*/*why*/*what*, and the answers are expected to be more comprehensive and cover multiple aspects.

Due to the lack of gold document annotations on ELI5, we adopt two alternative reordering strategies—rotational reordering and random shuffling. In rotational reordering for k documents, we first randomly initialize a document sequence $d_1, d_2, \cdots, d_k$, where d_i is the i-th document in the sequence. The input context is constructed by concatenating the documents in the order of $[d_1, d_2, \cdots, d_k]$, $[d_2, d_3, \cdots, d_k, d_1]$, $[d_3, \cdots, d_k, d_1, d_2]$, etc. Thus, rotational reordering enables all documents to appear at each position. In random shuffling for k documents, we randomly shuffle the document set k (i.e., same as the number of documents) times and obtain k document sequences for consistency.

Given multiple prompts for a question, among which only the document orders in the context are different, we calculate the average performance of the prompts with the highest and lowest $\log p$(question) for each question in the same fashion as described in Section 4.3.1 for NQ-Open.

[15] For further discussion, see Appendix D.1.

Results in Table 2 show that LMs achieve higher answer accuracy on the prompts with the highest $\log p(\text{question})$, compared with the prompts with the lowest $\log p(\text{question})$. This indicates LMs can better answer questions with higher question likelihood through document shuffling, demonstrating the strong instance-level correlation between $\log p(\text{question})$ with answer accuracy.

5 Improving Accuracy via Document Reordering

In Section 4.3, we find that $\log p(\text{question})$ strongly correlates with model performance at the instance level. I.e., given the same question, prompts with higher $\log p(\text{question})$ are likely to lead to higher answer accuracy on QA tasks. Therefore, we propose two methods for improving LM's performance through document reordering—naïve likelihood-based selection and gold document reordering. The experiment is conducted on the ELI5 dataset and a subset of 500 questions from NQ-Open, using `Mistral-7B-Inst-v0.3`, `LLaMA-3.1-8B`, and `LLaMA-3.1-8B-Inst`. Each question is associated with 10 and 20 retrieved documents, respectively, on ELI5 and NQ-Open.

5.1 Method 1: Naïve Likelihood-Based Reordering

Section 4.3 has shown that for each instance, the prompt that achieves the highest question likelihood during document shuffling likely leads to superior answer accuracy on QA tasks. We select the prompt that achieves the highest question likelihood during document shuffling as the LM input for each question. We consider this the most straightforward likelihood-based method for boosting answer accuracy.

5.2 Method 2: Gold Document Reordering

As observed in Figure 4, answer accuracy is the highest when the gold document is placed at the beginning of the document sequence. This suggests that answer accuracy can be improved by moving the gold documents towards the beginning of the context.[16] To this end, we propose to measure each document's relevance in answering the question through question likelihood. Using such means, we aim to detect the most relevant document and manually place it at the beginning of the document sequence to get an "optimal prompt".

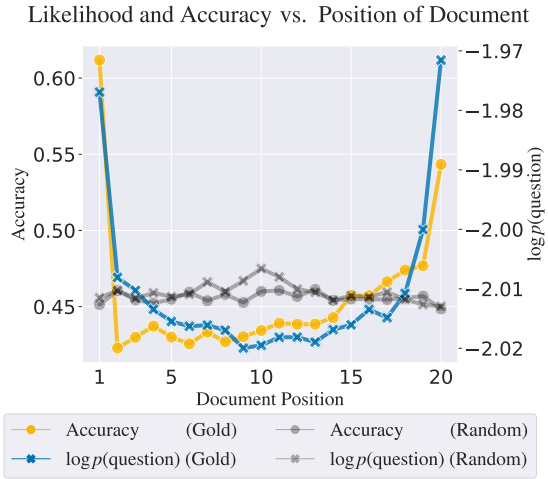


Figure 5: When the position of the gold document changes, both $\log p(\text{question})$ and accuracy curves are U-shaped. In contrast, both curves are flat for non-gold (denoted by random) documents.

5.2.1 Gold vs. Non-Gold Document

We have found in Section 4.3 that question likelihoods tend to be higher when the gold document is placed at the beginning or the end of the document sequence. When the gold document is placed in the middle of the context with non-gold documents at the beginning or end, question likelihood drops.

Besides, we also investigate the question likelihoods when changing the position of non-gold documents. Compared to the convex curve of the gold documents in Figure 4, the trend of $\log p(\text{question})$ during non-gold document shuffling is relatively flat, as illustrated in Figure 5.

5.2.2 Reordering via Relevance Identification

Based on the findings in Section 5.2.1, we propose ConvexScore to measure the relevance of each document with the question, defined as

$$\text{ConvexScore}_d = \text{InitScore}_d + \text{Convex}_d, \quad (1)$$

where InitScore_d is the sum of question log-likelihoods when placing the document d at the beginning and ending position of the document sequence with the rest documents placed in random order. Convex_d measures the curvature of the question likelihoods when the document d is placed at

[16] Although the performance also rises when the gold document is placed close to the end of context on LMs such as `Llama-3-8B`, it is nevertheless inferior to when the gold document is placed at the beginning. Besides, placing the gold document closed to the end does not necessarily lead to increased performance on many instruction-finetuned LMs (see Figure 4b). This finding is consistent with previous work (Liu et al., 2024; Chen et al., 2024).

each position, defined as below

$$\text{Convex}_d = \sum_{i=2}^{k-1}(\tilde{\ell}_{d,i+1} - \tilde{\ell}_{d,i}) - (\tilde{\ell}_{d,i} - \tilde{\ell}_{d,i-1})$$
$$= \tilde{\ell}_{d,1} + \tilde{\ell}_{d,k} - 2 \times \sum_{i=2}^{k-1} \tilde{\ell}_{d,i} \quad (2)$$

where $\tilde{\ell}_{d,i}$ is the question likelihood when the document d is placed at position i and the rest documents are placed in random order.

Therefore, the ConvexScore is calculated as:

$$\text{ConvexScore}_d = 2 \times (\tilde{\ell}_{d,1} + \tilde{\ell}_{d,k} - \sum_{i=2}^{k-1} \tilde{\ell}_{d,i}) \quad (3)$$

The document with the highest ConvexScore is moved to the beginning of the context to construct an optimized prompt for eliciting better answers.

5.3 Results and Analysis

Shown in Table 3, both naïve likelihood-based selection and gold document reordering can boost answer accuracy. On NQ-Open, where only one document in the sequence is relevant to the question, gold document reordering significantly improves the answer accuracy and narrows the gap to the upper bound. Furthermore, on the more challenging and practical QA benchmark ELI5, we also observe a modest improvement in answer accuracy, indicating that improving question likelihoods via document reordering can effectively obtain better LM responses.

Regarding efficiency, our proposed methods are mildly time-dependent thanks to the parallelizable computation of question likelihoods, where only the LM encoding module is used, with no reliance on LM decoding.[17] In our experiments, the average runtime of decoding an instance on ELI5 is 10 seconds, while it only takes an extra 0.8 seconds and 2 seconds, respectively, to encode the input prompt of naïve likelihood-based selection and gold document reordering. Compared with heuristic prompt engineering which requires whole decoding for judging the prompt quality (e.g. an extra 10 seconds for decoding for two candidate prompts), our likelihood-based methods marginally increase the computational cost.

[17] LM decoding (i.e., generation) requires a runtime approximately proportional to the number of generated tokens. Empirically, the only extra computational time for our methods is on the encoding phase for calculating likelihoods, so the overall runtime is the vanilla $\text{Runtime}_{\text{LM}}$ plus *one* extra LM going through, which is equivalent to generating the response with *one* additional token.

Model	Baseline	Likelihood Based	Gold Document	Upper Bound
NQ-Open (Answer Accuracy)				
`Mistral`	62.89	65.18	**65.72**	69.24
`LLaMA-3.1`	47.74	51.29	**51.36**	66.88
`LLaMA-3.1-Inst`	61.49	63.34	**63.56**	66.35
ELI5 (Answer Accuracy)				
`Mistral`	15.35	**15.63**	15.40	-
`LLaMA-3.1`	12.61	12.73	**13.33**	-
`LLaMA-3.1-Inst`	16.14	**16.90**	16.83	-

Table 3: Performance of our methods on NQ-Open and ELI5, the number of documents k is set to 20 and 10, respectively. `Mistral`, `LLaMA` and `LLaMA-Inst` stands for `Mistral-7B-Inst-v0.3`, `LLaMA-3.1-8B` and `LLaMA-3.1-8B-Inst` respectively. Baseline refers to the mean performance over k random document shuffling on each instance. The upper bound on NQ-Open is calculated as the performance when positioning the gold document at the beginning of the document sequence, which is not applicable for ELI5 since no gold document is marked in this practical dataset.

In summary, both proposed methods are effective and efficient. Although the improvement on ELI5 is relatively marginal compared to that on NQ-Open, given the more challenging nature of long answers and no specified gold document on ELI5, it still indicates that optimizing prompts with $\log p(\text{question})$ as a gauge is a promising direction.

6 Conclusion

In this work, we analyze the relationship between likelihood and language model's accuracy in solving tasks under the retrieval-augmented generation framework. Through experiments, we demonstrate that the question likelihood is affected by the order of documents in the input context. We reveal the correlation between question likelihood and answer accuracy at both the corpus level and instance level. Our findings show that it is possible to use likelihood to gauge language model performance and improve the quality of input prompts. We propose two practical methods for prompt optimization based on these findings. Experimental results show that both effectively and efficiently improve LM's accuracy on QA tasks, demonstrating that using $\log p(\text{question})$ as a gauge for optimizing prompts is a promising direction. We leave other prompt modification choices, beyond document reordering, for future study.

Limitations

One major limitation of our work is that only open-source LMs are studied in this work since we need full access to the question likelihoods. One possible way to extend our findings to closed LMs is to encode the prompts with the open-source LMs belonging to the same LM family. For example, it is possible to use `GPT-3` for computing the question likelihood and improve the input prompts for `GPT-4`.

Besides, regarding the types of prompt modification, we follow the experimental setup of previous works and, thus, mainly focus on document reordering in this work. Nevertheless, other prompt modifications may also contribute to obtaining a higher $\log p(\text{question})$ and improve answer accuracy. Considering that in this work we are taking the first step towards exploring the feasibility of prompt optimization without LM decoding, proving our hypothesis, and managing to optimize prompts with our findings, we leave other prompt optimizations for future study.

References

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021. One question answering model for many languages with cross-lingual dense passage retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 7547–7560. Curran Associates, Inc.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA. MIT Press.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Xinyi Chen, Baohao Liao, Jirui Qi, Panagiotis Eustratiadis, Christof Monz, Arianna Bisazza, and Maarten de Rijke. 2024. The SIFo benchmark: Investigating the sequential instruction following ability of large language models. *arXiv preprint arXiv:2406.19999*.

Ryan Cotterell, Anej Svete, Clara Meister, Tianyu Liu, and Li Du. 2024. Formal aspects of language modeling. *Preprint*, arXiv:2311.04329.

Sabit Ekin. 2023. Prompt engineering for chatgpt: a quick guide to techniques, tips, and best practices. *Authorea Preprints*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Louie Giray. 2023. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2024. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Lijia Ma, Xingchen Xu, and Yong Tan. 2024. Crafting knowledge: Exploring the creative mechanisms of chat-based search engines. *arXiv preprint arXiv:2402.19421*.

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Karl Pearson. 1895. VII. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240 – 242.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. 2021. The web is your oyster-knowledge-intensive nlp against a very large web corpus. *arXiv preprint arXiv:2112.09924*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. 2024. Model internals-based answer attribution for trustworthy retrieval-augmented generation. *arXiv preprint arXiv:2406.13663*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

C. Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 193–210, Santa Clara, CA. USENIX Association.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Illustration of Evaluation Metrics

The evaluation metrics for NQ-Open and ELI5 are illustrated with two examples in Figure 6.

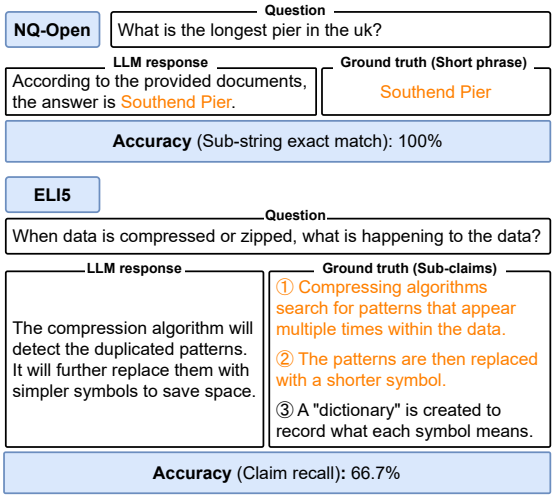


Figure 6: Evaluation metrics used in our experiments. On NQ-Open, the evaluation metric is exact string match. On ELI5, a pretrained NLI model is used to evaluate whether the LM output entails the reference claims.

B Prompt Templates

The prompt templates used for our experiments are given in Figures 7 and 8.

```
Write a high-quality answer for the
given question using only the provided
search results (some of which might be
irrelevant).

Document [1](Title: Southend Pier)
Southend Pier is a major landmark in . . .

Document [2](Title: Llandudno Pier)
Llandudno Pier Llandudno Pier is a Grade
II* listed pier in the seaside resort of
Llandudno. . .

Document [3](Title: Garth Pier) Garth Pier
Garth Pier is a Grade II listed structure
in Bangor. . .

. . .

Question: what is the longest pier in the uk

According to the provided documents, the
answer is Southend Pier.
```

Figure 7: An example prompt and LM output of multi-document question answering task. The prompt comprises (1) an instruction that describes the task to be solved, (2) a context that contains the information for solving the task, in which the gold document contains the ground truth answer, and (3) a question that describes the specific query. At the end of the prompt, we append an exemplar output that gives the ground truth answer to the question for evaluating the likelihood of the answer.

C Full Results on NQ-Open

We show the full results on NQ-Open in Figures 11–14.

D Discussion and Analysis

D.1 How does document order affect answer accuracy?

The objective of language modeling is to estimate a distribution over the sequence of tokens in a training set. When a sequence is assigned high likelihood, sequences that are similar to it, either semantically or structurally, are generally more frequent in the training data. Due to the exposure to real data during training, the LM is biased to only perform well on examples that are similar to ground-truth history distribution (**exposure bias**, Bengio et al., 2015; Ranzato et al., 2016; Cotterell et al., 2024).

In our experiments, we found that $\log p(\text{question})$ is much higher when the document carrying useful information is placed at the first and last place in the document sequence; $\log p(\text{question})$ is significantly lower when relevant information to the question appears in the middle of input context.

Thus, we hypothesize that the fluctuation of model performance is a consequence of the pretraining data where questions are more often asked about the very beginning or end of input contexts. This explains the trend of $\log p(\text{question})$ in Figure 4. Given a higher $\log p(\text{question})$, the model is less surprised about the prompt and the relevant answer can obtain a higher likelihood, contributing to a better generated response.

On the other hand, the reduction in task performance may result from low training data frequency, where the question likelihoods are relatively lower.

D.2 Instruction-Finetuned vs. Base Models.

In our analysis, we find base LMs, e.g., `LLaMA-3-8B`, tend to be more vulnerable to the changes in the position of the gold document. Their $\log p(\text{question})$, $\log p(\text{answer})$, and answer accuracy drop significantly when the gold docu-

```
Instruction: Write an accurate, engaging,
and concise answer for the given question
using only the provided search results
(some of which might be irrelevant). Use
an unbiased and journalistic tone.

Document [1](Title: Trash Islands - the
Ocean Garbage Patch): Trash Islands
Trash Islands of the Pacific and Atlantic
Oceans. . .

Document [2](Title: Where does our
garbage go? - Sea Turtle Camp): Pacific
Garbage Patch Landfills are a common human
solution for disposing of trash on land. . .

Document [3](Title: Plastic pollution
crisis: How waste ends up in our oceans -
Y108): our ecosystems as a whole. Plastic
is non-biodegradable. Every year, about
8-million tons of plastic. . .

. . .

Question: how does so much of our trash
end up in the ocean?

According to various sources, a significant
portion of the world's trash ends up
in the ocean due to a combination of
factors. While it's often. . . individuals
is necessary to mitigate the problem of
plastic pollution in the world.
[Answer length: 242 words]
```

Figure 8: An example prompt and LM output on ELI5. The prompt comprises (1) an instruction that describes the task to be solved, consistent with previous works on ELI5 (Gao et al., 2023; Qi et al., 2024), (2) a context that contains the information for solving the task, but *no gold document* is marked, and (3) a question that describes the specific query. At the end of the prompt, we append an exemplar output that gives the ground truth answer to the question for evaluating the likelihood of the answer.

ment is placed in the middle of the document sequence. On the other hand, the *performance* of instruction-tuning models is more robust to the positional bias caused by the changes in gold document positions as shown in Figure 4. Specifically, the $\log p(\text{question})$ still appears in a U-shaped curve, but the drop in answer accuracy and $\log p(\text{answer})$ is less significant for the instruction-finetuned model when the gold document is positioned at the middle.

The similar U-shaped trend of the $\log p(\text{question})$ curves of both base and instruction-finetuned models implies that question likelihood is affected little by the instruction tuning, which supports our claim that question likelihood reflects the distribution of pre-training data; while instruction tuning targets at the decoding phase of LMs, promoting the LM to assign the correct answer with a higher probability, which further contributes to improving LM accuracy, as shown by the highly simultaneous change of $\log p(\text{answer})$ curve and accuracy curve.

D.3 Context Likelihood Analysis

This work explores the correlation between question likelihood and answer accuracy when documents are in various orders. However, the likelihood for the document sequence, namely $\log p(\text{context})$, may also vary during the shuffling. To alleviate the concern that context likelihoods may affect answer accuracy, we compute the $\log p(\text{context})$ for questions in NQ-Open, when placing the gold document at different positions. To our observation, the average $\log p(\text{context})$ is approximately 1/10 compared with average $\log p(\text{question})$ per token, indicating that the context likelihood has a marginal effect on the likelihood at the input side. We investigate context likelihood in Appendix E for a comprehensive study.

E Context likelihood

In our experiments in Section 3, the context comprises the retrieved documents provided to assist the language model answering the question, which follows the context in the prompt templates.

E.1 Context Following Question

For a comprehensive study, we also explore the changes in $\log p(\text{context})$ when switching the position of context and question, i.e., putting the retrieved documents after the question.

We calculate two values relevant to an LM's input and output, namely the context likelihood $\log p(\text{context})$ of the input documents and the answer likelihood $\log p(\text{answer})$. When analyzing $\log p(\text{context})$, we switched to another prompt template in which the question precedes the documents. For more details, see Figure 10.

There are two common types of prompt templates used to construct inputs for LMs (Gao et al., 2023; Liu et al., 2024), where the question could either be placed before or after the retrieved documents. We show two examples in Figures 7 and 10.

Due to the autoregressive nature of decoder-only LMs, in a prompt template where the question follows the input contexts, e.g., the example

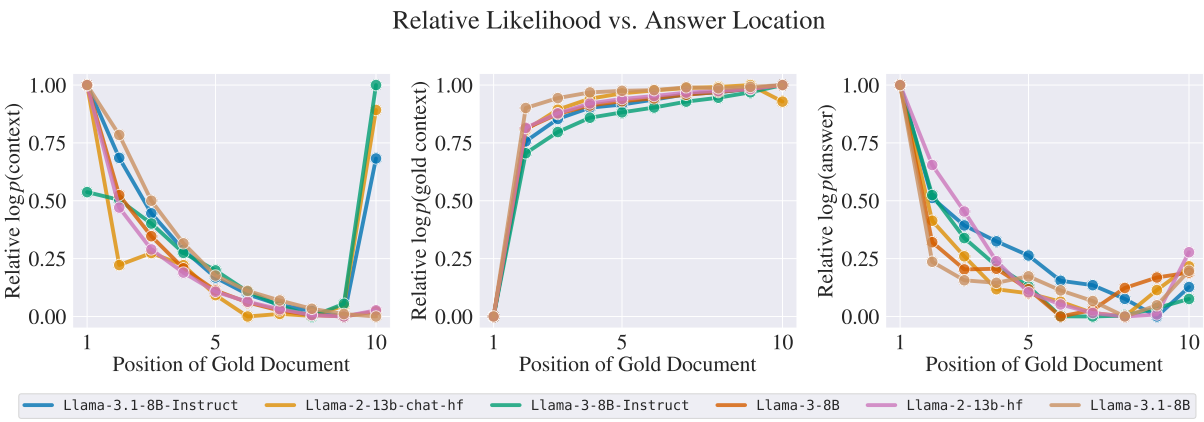


Figure 9: Relative log-likelihoods on 10 docs evaluated on LLaMA models using the alternative prompt template (Figure 10) where context follows question.

```
Write a high-quality answer for the
given question using only the provided
search results (some of which might be
irrelevant).
Question: what is the longest pier in the uk

Document [1](Title: Southend Pier)
Southend Pier is a major landmark in . . .

Document [2](Title: Llandudno Pier)
Llandudno Pier Llandudno Pier is a Grade
II* listed pier in the seaside resort of
Llandudno. . .

Document [3](Title: Garth Pier) Garth Pier
Garth Pier is a Grade II listed structure
in Bangor. . .

. . .

According to the provided documents, the
answer is Southend Pier.
```

Figure 10: An alternative prompt template in which the context comes after the question. The context log-likelihood $\log p(\text{context})$ is computed with this template.

in Figure 7, the likelihood of context remains unaffected by changes in question. Thus, we use an alternative prompt template, illustrated in Figure 10, to analyze the change in $\log p(\text{context})$.

In Figure 9, we plot the curves of $\log p(\text{context})$ and $\log p(\text{context})$. We find that on instruction-finetuned models, the context log-likelihood $\log p(\text{context})$ can also reflect the lost-in-the-middle effect. Meanwhile, on base models, $\log p(\text{context})$ tends to decrease monotonically.

To account for this phenomenon, we find that on instruction-finetuned LMs, the log-likelihood of documents drops significantly when the documents appear after the gold document. I.e., the mean likelihood of documents that appear after the gold document is much lower than the ones that appear before the gold document. When the gold document is located at the end of the context, the decrease in log-likelihood is minimized and $\log p(\text{context})$ increases, resulting in an increased likelihood when the gold document is placed at the end of the context.

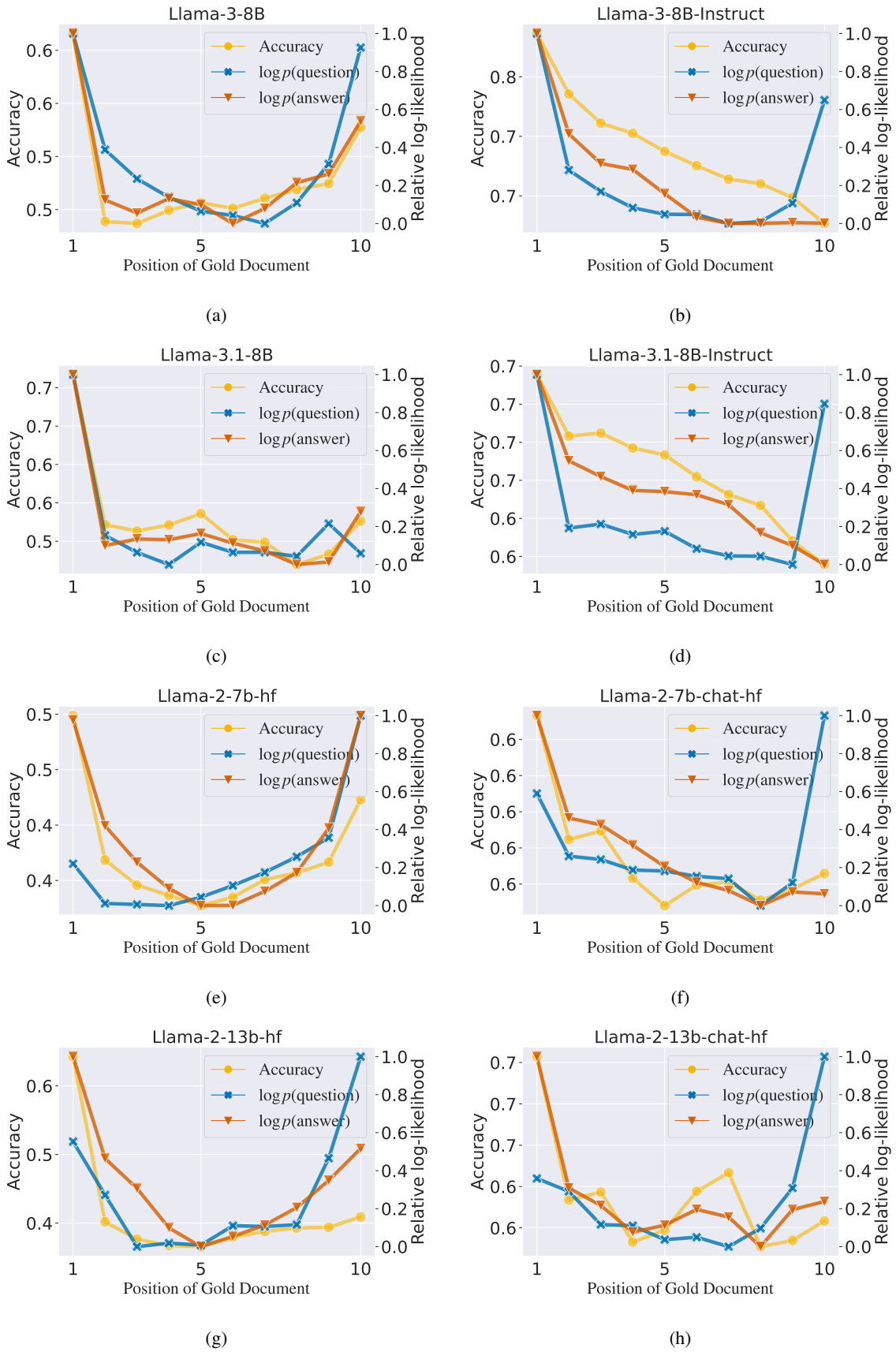
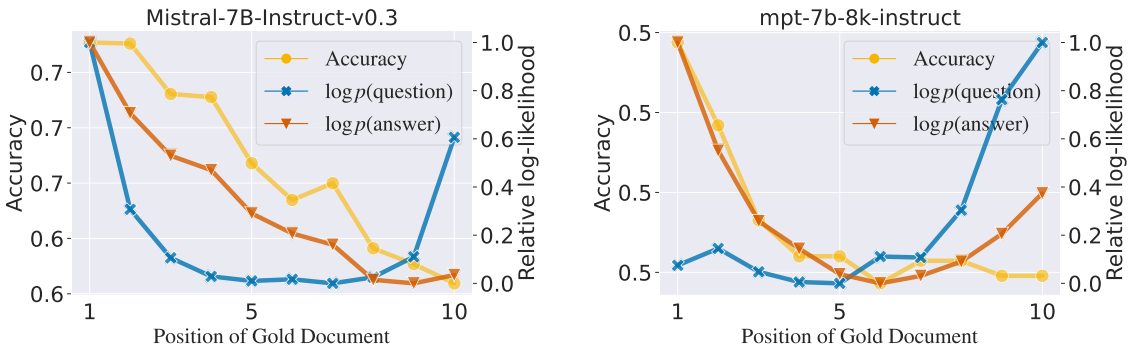


Figure 11: Relative question and answer log-likelihoods on 10 docs.



(a) (b)

Figure 12: Relative question and answer log-likelihoods on 10 docs.

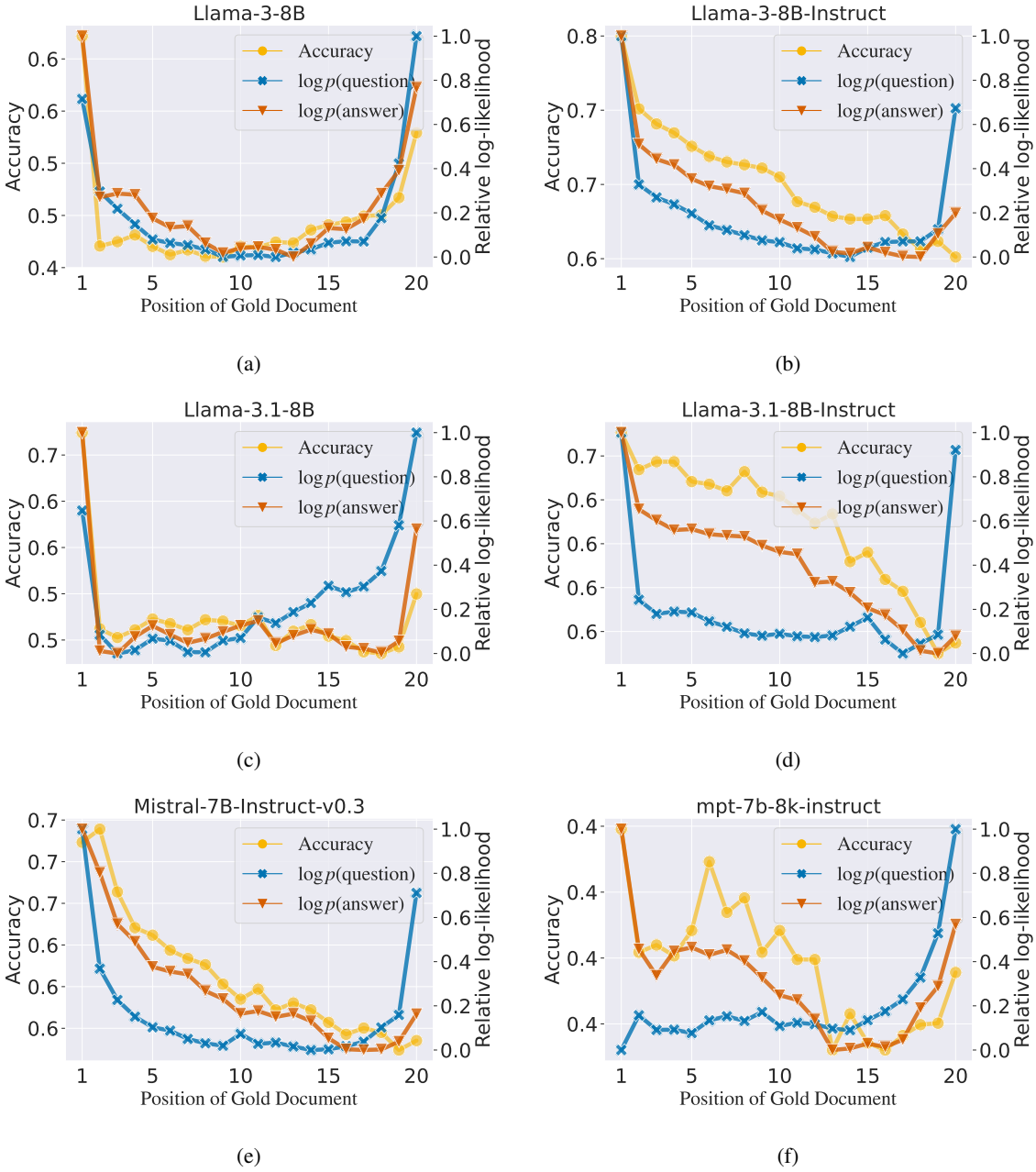


Figure 13: Relative question and answer log-likelihoods on 20 docs.

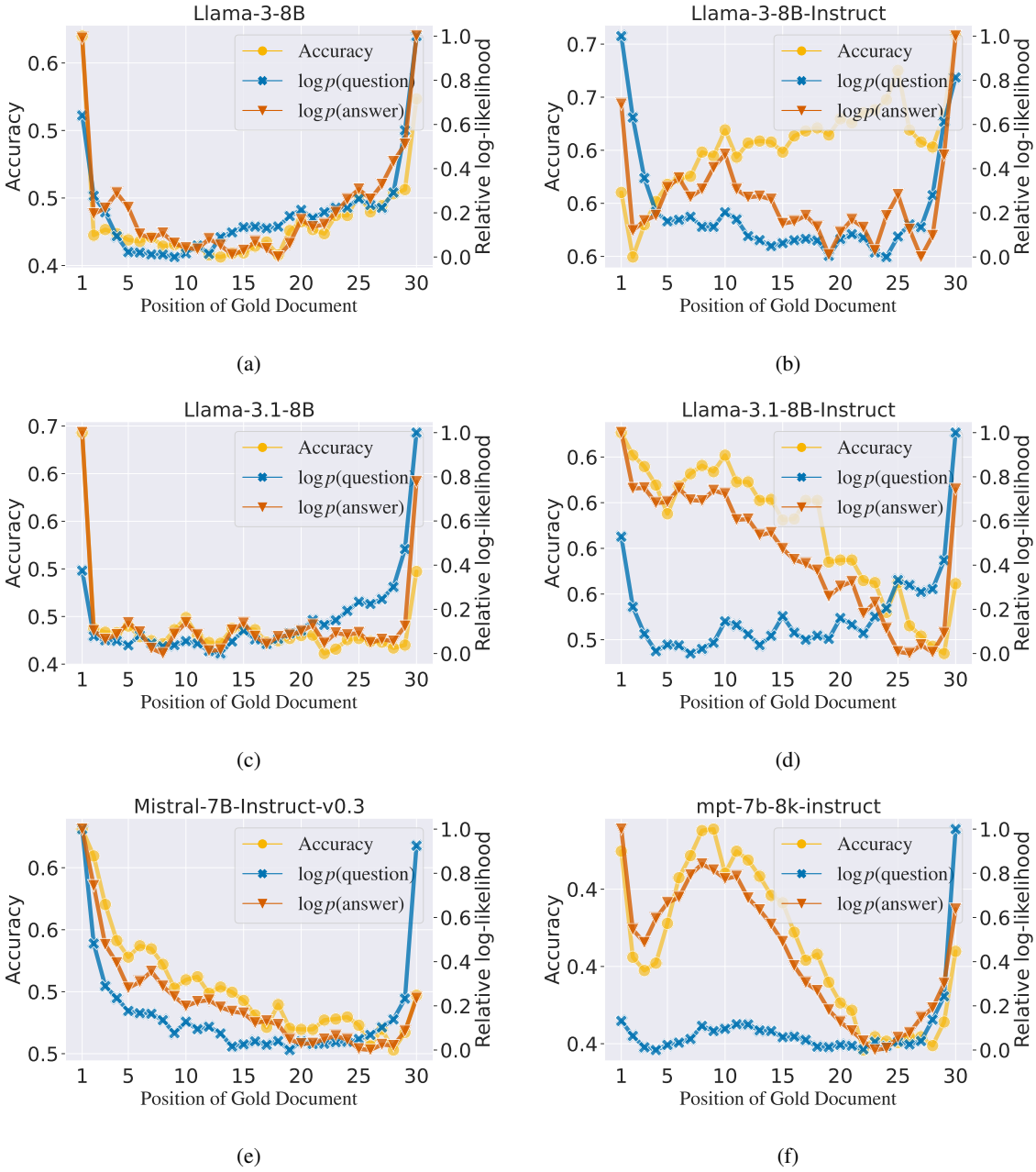


Figure 14: Relative question and answer log-likelihoods on 30 docs.