Analyzing and Improving Coherence of Large Language Models in Question Answering

Ivano Lauriola
Amazon AGI
lauivano@amazon.com

Stefano Campese
Amazon AGI
University of Trento
campeses@amazon.com

Alessandro Moschitti
Amazon AGI
amosch@amazon.com

Abstract

Large language models (LLMs) have recently revolutionized natural language processing. These models, however, often suffer from instability or lack of coherence, that is the ability of the models to generate semantically equivalent outputs when receiving diverse yet semantically equivalent input variations. In this work, we analyze the behavior of multiple LLMs, including Mixtral-8x7B, Llama2-70b, Smaug-72b, and Phi-3, when dealing with multiple lexical variations of the same info-seeking questions. Our results suggest that various LLMs struggle to consistently answer diverse equivalent queries. To address this issue, we show how redundant information encoded as a prompt can increase the coherence of these models. In addition, we introduce a Retrieval-Augmented Generation (RAG) technique that supplements LLMs with the top-k most similar questions from a question retrieval engine. This knowledge-augmentation leads to 4-8 percentage point improvement in end-to-end performance in factual question answering tasks. These findings underscore the need to enhance LLM stability and coherence through semantic awareness.

1 Introduction

LLMs have revolutionized the majority of Natural Language Processing tasks in the last years, including Question Answering (QA) (Li et al., 2024), chatbot (Achiam et al., 2023), coding (Nam et al., 2024; Ugare et al., 2024), and summarization (Jin et al., 2024) to name a few.

One of their main limitations is the stability of their output when slightly changing the input. This is reported in several previous works (Hu et al., 2024), especially in the prompt engineering research (Chen et al., 2024a; Cain, 2024). Although these works clearly point out the problem, understanding its causes is rather challenging, as the stability of the model, in addition to the fine-tuning step, also depends on pre-training. The latter involves the usage of a huge amount of data, which is typically different in different LLMs, e.g., using different training techniques, and parameters. This has forced researchers to study the stability of models using a black-box approach: trying different prompts, mostly guided by human cognitive considerations, and observing the impact on the single output or overall performance.

In this work, we analyze the coherence of multiple LLMs, including Mixtral-8x7B (Jiang et al., 2024), Llama2-70b (Touvron et al., 2023), Phi3-mini (3.8B) (Abdin et al., 2024), Smaug-72b (Pal et al., 2024), in multiple factual QA tasks. Specifically, we show that large models fail to provide the same or similar answers for semantically equivalent questions as input. We conjecture that the instability, or lack of coherence, is a symptom of the model not being able to fully understand a request, i.e., accessing its parametric knowledge necessary for generating a correct answer.

We show that redundant information encoded in the prompt, as multiple equivalent variations of the same question, can increase the accuracy of the model and its coherence, thus mitigating the understanding issue. Intuitively, LLMs can exploit different semantic patterns from the equivalent questions to better connect the user's request with their parametric knowledge.

We exploit this finding to design a principled approach to improve LLM coherence and accuracy, leading to greater stability through Retrieval Augmented Generation (RAG) (Gao et al., 2023; Chen et al., 2024b). In short, we built (i) an index of 38M of questions, and (ii) a dense retrieval engine, which, given a target question, retrieves semantically equivalent (or similar) queries. Given a question asked to a general LLM, our system first, retrieves k similar questions and then provides them to the LLM along with the original request. In this work, we refer to this approach as question-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 11740–11755

RAG (q-RAG). Our results show that all analyzed LLMs significantly benefit from our approach, with an improvement in both accuracy and coherence, measured as the semantic similarity of answers for equivalent questions.

2 Related Work

Several LLMs of diverse size, ranging from a few to hundreds of billion parameters, pre-trained on web-scale corpora, have been recently introduced, e.g.: GPT family (Achiam et al., 2023; Brown et al., 2020; Radford et al., 2018), Llama (Touvron et al., 2023), Mixtral (Jiang et al., 2024), Smaug (Pal et al., 2024), Falcon (Almazrouei et al., 2023), or Phi3-mini (Abdin et al., 2024). These models can be successfully applied to various NLP tasks, e.g., Question Answering (Li et al., 2024), chatbot (Achiam et al., 2023), coding (Nam et al., 2024; Ugare et al., 2024), and summarization (Jin et al., 2024), achieving high accuracy. However, their parametric knowledge shows some limitations, e.g., Basmov et al. (2024) showed that LLMs, applied to Machine Reading tasks, may easily fail when the input context is not aligned with the internal knowledge. Similar mechanisms produces lack of coherence of their results. For instance, Zheng et al. (2023) showed that LLMs are vulnerable to option position changes in multiple-choice QA tasks. Raina et al. (2024) pointed out LLM weaknesses with respect to adversarial attacks, e.g., when attempting to manipulate the output. Finally, Chatterjee et al. (2024) introduced a measure to quantify the sensitivity of an answer for a given prompt.

Our analysis of coherence is different from previous work as it is based on observing LLM output when semantically equivalent questions are used in their input. Moreover, we provide a principle approaches to mitigate this issue.

Ella Rabinovich (2023) used equivalent questions in LLMs for a different purpose: they introduced PopQA-TP, a meticulously curated dataset of 118K high-quality paraphrased questions. PopQA-TP expands the original PopQA (Mallen et al., 2022), creating paraphrasing of each of the 14K initial questions. PopQA-TP can be used for benchmarking LLMs' ability of maintaining semantic consistency across variations of the same question. However, the authors used PopQA-TP for designing an automatic evaluation model. This predicts the LLM's likelihood of answering a question correctly, using semantic consistency metrics, i.e., cosine similarity between answer embeddings, with other predictors, such as question subject popularity and answer certainty.

Retrieval Augmented Generation Keeping LLMs' knowledge up-to-date and covering niche information is an open research question, as it requires additional training, which is computationally expensive and can cause the forgetting of existing information. Retrieval Augmented Generation (RAG), e.g., (Gao et al., 2023; Chen et al., 2024b), aims to address the challenges above by retrieving external knowledge from large and updated sources, e.g., the Web (Lewis et al., 2021a). RAG techniques have been shown to increase the performance of LLMs in various tasks, including Question Answering (Siriwardhana et al., 2023), Answer Selection (Gabburo et al., 2022), and clinical medicine (Zakka et al., 2024).

Prompt engineering Prompt engineering is the task of crafting specific input (e.g. instructions) to guide LLMs' outputs, ensuring the generation of accurate responses, tailored to the target application. Prompt engineering introduce flexibility in modeling a task, thus reducing the need of fine-tuning LLMs on specialized tasks (Wei et al., 2022a). Radford et al. (2019) introduced the idea of *fine-tuning with minimal task-specific modifications* through prompts, demonstrating its effectiveness in various NLP tasks. Jiang et al. (2020) proposed *language model probing*, which creates suggestions for obtaining specific model behaviours, improving the understanding of its capabilities and limitations. Similarly, Wei et al. (2022a) showed that instructions can improve zero-shot performance on QA, reasoning, and story generation. Other studies have demonstrated that prompt engineering can be as effective as hundreds of training data points (Scao and Rush, 2021).

Prompting can be used to affect other qualitative aspects of generation, e.g., Wallace et al. (2021) used prompts to mitigate unwanted biases while improving fairness. More recent prompting approaches improve agents' answer quality, e.g., Chain-of-thought prompting entails breaking down the question and reasoning over possible solutions before generating the answer, both in zero- (Kojima et al., 2023) or few-shot settings (Wei et al., 2023). Additionally, graph prompt methods have been utilized (Yao et al., 2023).

3 Question-answer coherence

We consider question-answer coherence as a proxy to quantify the ability of a LLM of retrieving the knowledge necessary to correctly answer a question. Intuitively, if a model can correctly answer a specific question, e.g., *What is the capital of Italy?*, it can also answer other semantically equivalent questions, e.g., *Can you tell me what's the capital of Italy?*, or *name of the city that serves as the capital of Italy*. Conversely, if a model fails to answer a question, it will likely expected to fail equivalent questions, since it lacks the necessary knowledge. A different behaviour of the model indicates an underlying issue since failing to answer an equivalent question answered correctly suggests the model cannot access and use its knowledge.

Similarly to previous work (Ella Rabinovich, 2023), we measure the model coherence with respect a question, q, as the average similarity between the answers embeddings generated from a set of questions semantically equivalent to q. We can estimate model coherence on a set of different questions (each associated with different equivalence clusters) by averaging their coherence.

Formally, let $\mathcal{Q}$ be a set of open-domain well-formed questions and let $\mathcal{C} \subseteq \mathcal{Q}$ be a set (or *cluster*) of questions such that $\forall (q_i, q_j) \in \mathcal{C}^2 : q_i \equiv q_j$, where $\equiv$ indicates that two questions are semantically equivalent. We used the equivalence definition introduced by Campese et al. (2023): Two questions (q_i, q_j) are semantically equivalent iff they have the same information-seeking intent and their answers can be interchanged, more formally: $\forall_a : l(q_i, a) \leftrightarrow l(q_j, a)$, where l is a labeling function such that $l(q, a) = 1$ if the answer a is correct for q given a certain interpretation of correctness, -1 otherwise. Given a model δ, and a set of m clusters $\{\mathcal{C}_r\}_{r=1}^{m}$, such that $\mathcal{C}_r = \{q_1, .., q_n\}$, $q_i \equiv q_j \forall i, j$, the coherence is defined as

$$\sum_{r=1}^{m} \left(\frac{2}{mn(n-1)} \sum_{(q_i,q_j)\in\mathcal{C}_r^2,\ i<j} \langle \mathbf{e}(\delta(q_i)), \mathbf{e}(\delta(q_j)) \rangle \right),$$

where $\delta(q)$ is the δ-generated answer for q and $\mathbf{e}$ is a text-embedding model[1]. The higher the value, the higher the probability that the model answers two semantically equivalent questions coherently. Note that this approach is not suitable for multi-answer, e.g.: subjective, questions, where different correct answers may have very different embeddings. However, this study focuses non-subjective queries with well-defined and verifiable answers.

3.1 Question prompting

We propose a principled and generalized approach to build a prompt specifically designed to improve the question-answer coherence. We consider a two-step pipeline: First, we use an external system that, given an input query, provides k additional equivalent questions with different wording. We call these Support Questions (SQs). Then, we use a prompt that asks the model to generate an answer for the input query by looking at the SQs to disambiguate or clarify the original intent.

We implemented a state-of-the-art Question Retrieval System (QRS) following the approach by Campese et al. (2023) to find SQs. QRS consists of (i) a DataBase of questions with their correct answers and (ii) a dense retrieval model that queries the DataBase (or index) and returns the top k similar question/answer pairs.

Our DataBase consists of 38M question/answer pairs, including (i) 6M pairs from various public sources (WikiHow, Quora pairs, Natural Questions) and (ii) additional 32M pairs from Probably-Asked-Questions (Lewis et al., 2021b) (PAQ). PAQ is a large-scale collection of questions and answers automatically generated and designed to train generative QA models. Note that the 6M pairs from annotated resources are the same used by Campese et al. (2023), consisting of a mixture of annotated and generated pairs. The same authors estimated that the correctness of the answer with respect to the associated question is 93%. Differently, PAQ consists of automatically generated pairs without human annotations. The authors of the resource estimated a question/answer pair correctness of 83%. In order to increase the quality of the database and to reduce noise due to wrong answers, we ran the Answer Selector model from Di Liello et al. (2023). The model takes a question/answer pair as input and produces a score representing the likelihood of the answer being correct. We ran the selector model on all original 64M pairs from PAQ and selected only the 50% pairs with highest selector score. Our manual annotation of 200 QA pairs randomly sampled from selected PAQ pairs shows accuracy of 96

The second key component of the QRS pipeline is the dense retrieval model that queries the DataBase and selects the most similar questions.

[1]In our work, we used: `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

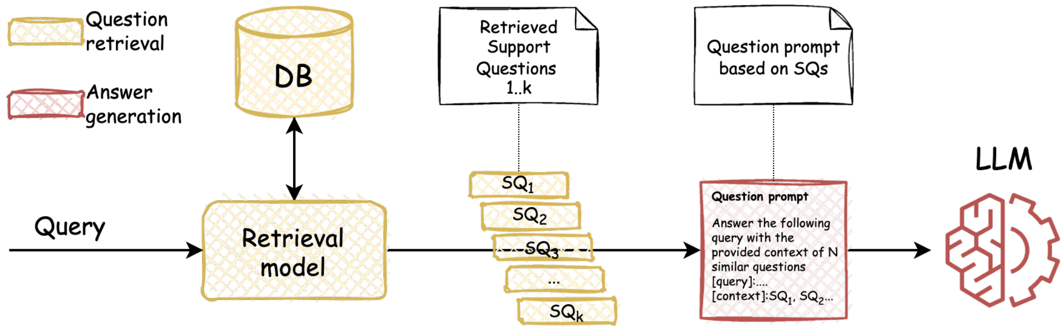

Figure 1: End-to-end QA pipeline. First, a question retrieval component finds similar questions from a pre-computed DataBase of question/answer pairs. Then, an LLM generates the answer while consuming the retrieved content.

We started from MiniLM-12L-v2[2], a bi-encoder of 33M learnable parameters. We continuously pre-trained the public checkpoint on a corpus of 900 million sentence pairs for semantic text similarity publicly available (Reimers and Gurevych, 2019), and we fine-tuned the resulting model on a dataset for question ranking task Campese et al. (2023) (Other training details are in Appendix B). This architecture allows us to efficiently return the k most similar stored questions (and their associated answers) based on similarity scores.

We fed the retrieved k SQs together with the original question into an LLM to generate an answer. We consider a prompt that emphasizes the task of answering the input query while considering the retrieved question/answer pairs as possible sources of information. The prompt may vary model by model, thus we ran an initial evaluation to find an optimal prompt for each model. The final prompt is shown in Appendix A.

The complete pipeline that includes question retrieval and answer generation steps is depicted in Figure 1. In the remainder of this article, we refer this strategy as *question prompting*. Note that the retrieval system used to collect SQs can be easily replaced by different solutions, e.g., paraphrasing models or other LLMs. However, given the cost of running LLMs, we focused on a simple QRS for cost/efficiency reasons. Alternative solutions are further analyzed and assessed in Section 4.4.

4 Experiments

We ran various experiments to evaluate the contribution of question prompting when applied to various popular LLMs, including Mixtral-8x7B (56B parameters), Llama2-70b (70B), Phi3-mini (3.8B), and Smaug-72b (72B). First, we evaluated the performance of the technique on two benchmarks for question equivalence. Then, we extended these findings to general QA tasks, showing the contribution of question prompting compared to classical RAG approaches. Next, we analyzed alternative LLM-based approaches to find SQs and to build the question prompt. Finally, we quantified the coherence of LLMs based on the definition provided by Ella Rabinovich (2023). In all experiments, we set the Temperature value to 0.001. We used 8xV100 32GB and Amazon Bedrock to run LLMs.

4.1 Datasets and metrics

We ran our experiments on multiple benchmarks, including:

Question Ranking (QR) - (Campese et al., 2023) is a dataset used to train question similarity/ranking models. It consists of 15,000 open domain queries. Each query is associated with 30 similar questions. The binary label for each query-question pair indicates whether the two are semantically equivalent or not, according to the definition described in Section 3. We considered test queries for which at least 5 equivalent questions exist (positive binary label). A query with the associated 5 questions defines a single cluster. For each cluster, we used the initial question as test query and the associated 5 questions as SQs. Overall, we extracted 762 clusters from the test split to be used in our experiments.

PopQA-TP - (Ella Rabinovich, 2023) is a large-scale open-domain resource consisting of 118k entity-centric QA pairs divided into 14k clusters of semantically equivalent variations. Different from QR, PopQA-TP consists of larger clusters. In this work, we considered clusters of dimension 10 or above. In all experiments involving this resource,

[2]https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

we considered 5 out of 10 questions from a cluster as SQs and the remaining 5 questions as test. Overall, we extracted 5518 clusters.

Open domain QA - To evaluate question prompting end-to-end, we sampled questions from multiple public open-domain QA datasets, including (i) Natural Questions (Kwiatkowski et al., 2019): questions from real Google users with associated answers from Wikipedia, (ii) Quora Question Pairs (Wang et al., 2020): user queries sampled from the homonym website, (iii) PAQ: generated questions, and (iv) TriviaQA (Joshi et al., 2017): a set of challenging questions authored by trivia enthusiasts.

Annotation - In all experiments involving QR and other open domain QA datasets, we used Amazon Mechanical Turk to evaluate the answers produced by LLMs with different prompts and configurations. For each question and LLM-generated answer pair, the annotators were tasked to indicate whether the answer was correct for the input question. We consider an answer to be correct if contains the exact information to answer the question. Annotators were asked to verify the information by using a search engine to compensate for any gaps in their knowledge and to provide a factual judgment. In addition, annotators were asked if the answers were natural or not. A natural answer is direct (i.e. it's not answering a similar question), precise (e.g. no additional unsolicited information), and fluent (e.g. no repetitions). In all evaluations, annotators were precluded from accessing SQs. Annotation guidelines and a deeper description of the annotation process are reported in Appendix C.

Differently, PopQA-TP evaluation is automatic. The dataset contains, for each question, a list of entity-based reference answers so an exact match can easily be applied to verify the generated answer's correctness.

4.2 Question equivalence benchmarks

Given QR and PopQA-TP clusters, we evaluated the accuracy of multiple LLMs, including Mixtral-8x7B, Llama2-70b, Phi3-mini, and Smaug-72b, to generate the answers with and without question prompting. For each cluster, we used 5 questions as SQs directly (no actual retrieval) in our prompt. The remaining questions, i.e. 1 per cluster in the case of QR and 5 for PopQA-TP, are used as test questions. Results on PopQA-TP are then averaged over the 5 input questions. This simplification allows us to observe the contribution of question prompting in a clean setting with semantically equivalent SQs manually annotated, without the complexities and noise of a full RAG setting. QR and PopQA-TP results are shown in Table 1.

Results on both datasets suggest that all models generally benefit from question prompt as their accuracy tend to increase. Note that, in the case of QR, the improvement holds whether considering naturalness or not as part of correctness. Unlike other models, we observed a drop in accuracy for Smaug-72b on QR. We conjecture that this behavior is strictly correlated to the model training. Smaug-72b specifically targets performance improvement on datasets, e.g. ARC (Clark et al., 2018) and HellaSwag (Zellers et al., 2019), designed for understanding tasks and considered difficult for most existing LLMs. Thus, adding other redundant information may not necessarily help. However, the same model improves +4.52% on PopQA-TP. We hypothesize this may depend on various aspects. First, the two datasets require completely different type of answers. PopQA-TP answers are entity-like and the model is asked to not generate a long text, which may differ from original training of the model. Second, QR is an older resource and Smaug may have been trained on some of those questions. Third, PopQA-TP is much more challenging than QR (see models' accuracy) and the dataset complexity may have a key role for question prompt contribution. See Appendix D for some examples of questions from QR where Smaug fails when using question prompt.

Note that the added information consists of manually curated semantically equivalent questions only, without any new knowledge.

4.3 Coherence evaluation

We measured the coherence of the models, as described in Section 3, on PopQA-TP when answering 5 different equivalent questions from the same cluster (while using the remaining 5 as SQs in our prompt). Note that the same procedure cannot be applied to QR as the latter has a single query per cluster available as test. Results are showed on Table 1. To emphasize, although LLama2 is the model with highest coherence, it is not the one with highest accuracy. This findings imply that coherence is a property of the model and it is not necessarily related to its performance. Intuitively, a coherent model should know how to answer all or none of the questions of the same cluster. By definition, a model that always provide the same

Model	QR: Correct		QR: C. & Natural		PopQA-TP: EM		PopQA-TP: Coher.	
	Base	Q. prompt	Base	Q. prompt	Base	Q. prompt	Base	Q. prompt
Mixtral-8x7B	78.48	81.10	40.29	46.95	16.72	20.3	53.21	81.21
Llama2-70b	77.69	84.38	54.20	62.73	15.69	17.37	81.36	84.51
Phi3-mini	68.76	71.78	54.46	58.53	5.01	5.14	43.46	61.71
Smaug-72b	83.59	72.31	68.77	57.21	13.89	18.41	54.51	75.97

Table 1: Models accuracy while using simple questions or question prompts on QR and PopQA-TP. In the case of QR, human labels are used defining answers to be correct and or natural. PopQA-TP evaluation is based on exact match applied on entity-answers from the original resource. The coherence on PopQA-TP is also displayed.

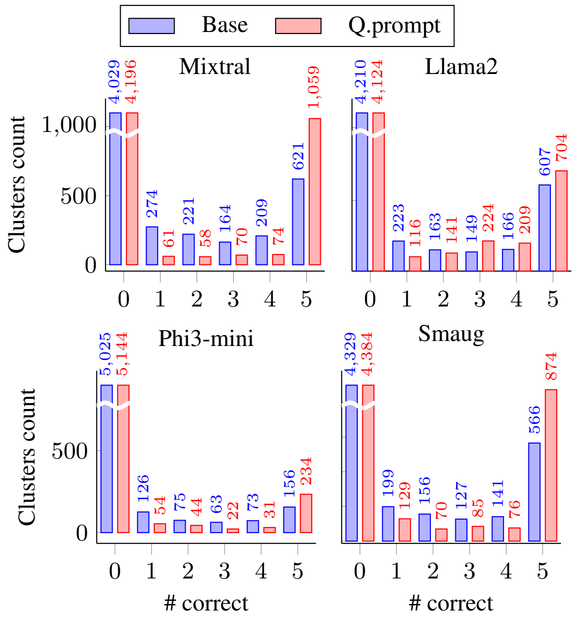

Figure 2: Number of correct answers per PopQA-TP cluster.

wrong answer is, intuitively, very coherent.

In order to understand the relation between accuracy and coherence, and based on the alternative intuition of the coherence above, we evaluated the following. For each PopQA-TP cluster of 5 test questions, we counted how many times a model generates a correct answer. 2 or 3 correct answers in a cluster of 5 equivalent questions may indicate lower coherence, whereas 0 or 5 indicates highest level of coherence, where the model knows the answer for all equivalent questions or it can't answer any. Note that this additional coherence measure has some blind-spots compared to the original embedding-based metric as it does not contemplate that two wrong answers can be semantically very distant (i.e. low coherence).

Figure 2 shows how many questions in each cluster are correctly answered. The plots show that question prompting moves the distribution away from the less coherent region (2-3 correct answers per cluster) to a more coherent section, consisting of more 0 and 5 correct answers per cluster. Notably, a shift towards 5/5 correct answers is expected as question prompt improves the accuracy, and thus the probability of increasing the number of correct answers per cluster. However, question prompt increased the amount of clusters with 0 correct answers for 3 out of 4 models. The only exception is Llama2 which, according to Table 1, is the model with highest semantic coherence.

This evidence suggests how question prompt is effectively improving the coherence of the model beyond mere accuracy.

4.4 End-to-end evaluation

The previous experiment showed how state-of-the-art LLMs are not robust to question reformulation, highlighting a possible understanding issue. However, the experiment relied on a curated benchmark with manually annotated SQs, which are not available in practical end-to-end QA use cases. In the following, we show how a RAG framework to retrieve SQs, introduced in Section 3.1, can mitigate the issue and improve the overall accuracy in end-to-end QA tasks.

We sampled 500 test queries from each Open Domain QA dataset reported in Section 4.1, for a total of 2000 queries. Given the complexity of the annotation process and the number of experiments to evaluate question prompting, we narrowed our assessment by considering the Mixtral model only. We consider the following configurations:

Base prompt - We directly asked the model to answer the input question without providing any additional information through a standard prompt for QA tasks.

Question prompt - We asked the model to answer the input question while observing top-k similar questions (SQs) retrieved by our auxiliary question search approach (QRS).

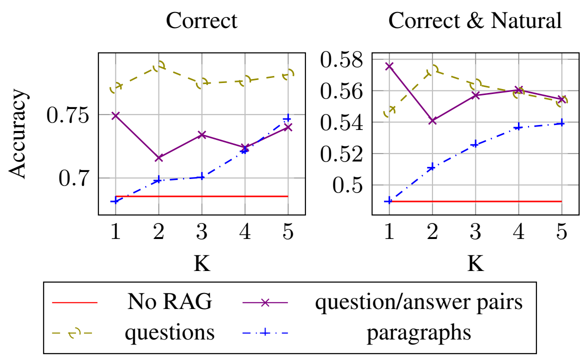

Figure 3: Left: Accuracy of Mixtral while using $k = 1 \ldots 5$ additional questions or question/answer pairs from our proposed question prompting, or 1 to 5 wikipedia paragraphs retrieved through BM25. Right: we consider unnatural answers as incorrect.

Q/A prompt - We extend the Question prompting by exposing the answers associated with the top-k similar questions. Note that, as described in Section 3.1, QRS stores and retrieves precomputed question/answer pairs. Thus, SQs can be paired with their answers in the prompt.

Paragraphs - We consider a classical RAG approach where we retrieve top-k paragraphs from a document index and include these paragraphs in the input. We used an index of Wikipedia paragraphs (Lin et al., 2021) queried through a DPR model[3].

All prompts used are reported in Appendix A. Figure 3 shows the accuracy of these configurations while increasing the value of k, i.e. the total amount of retrieved items (questions, q/a pairs, or paragraphs), from 1 to 5.

The plots highlight multiple key aspects. First, all RAG or prompt-based techniques improve over the trivial baseline where the model is asked to produce an answer given the simple question, without additional information. Second, the question prompting shows better improvement in accuracy compared to classical paragraph-based RAG. The gap between the two configurations increases when we require answers to be natural.

Note that SQs and paragraphs bring different types of knowledge and information. On the one hand, SQs used in question prompting are designed to be equivalent to the input query. QRS is explicitly trained to maximize the similarity between input query and retrieved questions. On the other hand, paragraphs are retrieved through classical document retrieval techniques designed to fulfill the input information-seeking request. In other words, SQs are designed to *shuffle* already available information to make it more appealing for the model, whereas paragraphs are meant to compensate for the lack of LLM knowledge. These considerations, in conjunction, suggest that LLMs may fail to understand the input query, and SQs may trigger mechanisms to retrieve and leverage the correct parametric knowledge.

Next, we observed that using question/answer pairs in our prompt further improves the correctness. This approach combines the advantages of question prompting and classical RAG-like methods. By incorporating SQs we help the model to understand the request and by adding actionable answers (sentences), we ingest information to compensate lack of parametric knowledge.

4.5 QRS and RAG performance

Previous results showed that question prompting improves over simple paragraph-based RAG. We further investigated to understand whether the improvement depends on the type of prompt itself or on the accuracy/recall of the information in the input. Table 2 shows the accuracy of the base systems (retrieval only, no generation involved) used to retrieve information for the LLM: QRS, i.e. the system used to build the question prompting, and RAG based on Wikipedia paragraphs retrieved through DPR. In the case of QRS, we evaluated the precomputed answer associated with the most similar question retrieved. We used the same 2000 test queries used in the previous experiment.

Metric	QRS	RAG
Correct	56.9	56.2
Correct & Natural	37.4	31.7

Table 2: Top-1 retrieved performance, QRS vs RAG.

In terms of mere correctness, QRS and RAG show similar standalone performance. We observed that QRS provides answers that are considered more natural compared to RAG (+5.7%). However, it is worth noticing that this is expected and it depends on the annotation guidelines and our definition of naturalness. Paragraphs typically contain more information and they don't answer a query directly, thus increasing the probability of being considered not natural. To this end, although

[3]https://huggingface.co/facebook/dpr-ctx_encoder-multiset-base

	No SQs	QRS	Q Gen.	CoT
Corr.	73.5	79.3	78.0	75.3
Corr. & N.	52.5	62.3	58.8	58.5

Table 3: Question Prompt performance with different methods to generate Support Queries. The DataBase approach (QRS) achieves better performance compared to other LLM-based SQs generation.

we report Correct & Natural accuracy for completeness, we focus on correctness only.

This result suggests that question prompting does not lead to better performance because the base system used to retrieve the information (QRS vs RAG) is more accurate.

4.6 Question generation

In previous experiments, we considered an external QRS system to provide SQs to be consumed in our prompt technique. However, question prompting is agnostic to the black-box support system that provides questions. We analyzed alternative approaches based on LLMs to generate questions. Specifically, we considered the following configurations:

No SQs - We run the LLM without SQs. We used a simple QA prompt.

QRS - Our approach queries a pre-computed DB of questions through question search models to find SQs.

Q Generation - we generate the answer in two steps. First, we ask the LLM to generate SQs with a simple prompt. Then, we use generated SQs in our question prompt, replacing QRS output, to generate an answer.

Chain-of-Thoughts - Inspired by CoT framework (Wei et al., 2022b; Kojima et al., 2024), we combine the two generation steps into a single pass. The prompt asks the model to virtually generate SQs and consume them to build an answer in a single step. We hypothesized that the model does not need explicit SQs if it can generate and include them as part of its reasoning.

Similarly to previous experiments, we used Mixtral-8x7B. We evaluated these strategies on 500 queries randomly sampled from NQ, Quora, PAQ, and TriviaQA. For all SQs-based configurations, we used k=5. Results are shown in Table 3.

Input: How old was jacquline wilson when her first book got published? **Generated:** What was the age of Jacqueline Wilson when she experienced the publication of her initial book? **Retrieved:** How old was jacqulien wilson when she wrote her first book?
Input: is it dangerous to eat expired yogurt? **Generated:** Is consuming out-of-date yogurt hazardous to one's health? **Retrieved:** How long after the expiration date is yogurt safe?
Input: How do you calculate dimensions? **Generated:** What is the method for determining dimensions? **Retrieved:** how do you work out a volume of a shape?

Table 4: Examples of generated and retrieved SQs.

Results show that the QRS approach achieves better performance compared to other LLM-based SQs generation techniques. To better understand this finding, we manually evaluated the semantic similarity of input queries with respect to 100 SQs generated by the LLM and 100 SQs retrieved through QRS. We estimated that 95% and 92% of generated and retrieved questions are semantically equivalent. Furthermore, we observed that generated questions mainly add some words or replace some words with synonyms. Differently, retrieved questions may expose additional facets relevant to answering the input query. See Table 4 for a few selected examples.

In the 1st and 2nd examples reported, retrieved questions may force the model to gather and consume more actionable information from its parametric knowledge. The retrieved question *How long after the expiration date is yogurt safe?* may help the model gather more detailed information compared to a simple *is it dangerous* Yes/No. We conjecture that retrieved SQs help the model to activate the parametric retrieval of relevant information that can be used directly or indirectly.

Note that there may be some room for improvement in SQs generation approach as we may use different specialized prompts, e.g. by asking the model to generate *similar* questions. However, beyond simple accuracy, QRS is much more efficient as, in our setting, it only requires at runtime (i) question embedding generation through a 33M parameters model, and (ii) querying a dense index of 38M 384-dimensional embeddings, which can be done in a few milliseconds with proper hardware. This is a more efficient solution than running a 56B parameters LLM (Mixtral-8x7B).

5 Conclusions

In this paper, we showed that popular LLMs have poor coherence when answering multiple lexical variations of the same questions. To mitigate the issue, we introduced a prompting technique that uses semantically equivalent questions, retrieved through a query retrieval framework, to improve the coherence the models. Multiple experiments on various benchmarks suggest that our approach improves both the coherence of the models and their accuracy.

Our work highlights the limitations of popular LLMs and sets foundations for future research on coherence improvement.

6 Limitations

Although multiple experiments described in this paper provide a strong indication of LLMs understanding ability and the benefits of Support Questions (SQs), results may be directional and there may be some limitations that need to be addressed in future work.

First, our analysis includes four different large language models (LLMs), and our results suggest that some of them may be more robust to question reformulation due to their high coherence. We considered models with up to 72 billion parameters, but our work lacks evidence for larger models such as Gemini or GPT-4. Second, LLM performance can be further boosted by playing with prompts. For instance, we observed that our question retrieval approach provided better SQs compared to LLM-based question generation. We conjectured that generated questions are too similar to each other and retrieved SQs may force the model to find additional information from its parametric knowledge that may indirectly help the main task. This evidence may indicate that other prompts that ask to generate relevant but not equivalent SQs may be beneficial in the end-to-end task. Then, this work mainly observes and emphasizes the coherence issue, providing a quick yet reliable solution. However, deeper analyses to understand the reasons behind the lack of coherence are needed. Finally, our work is based on a zero-shot setting. We hypothesize that models' coherence can be optimized through specialized fine-tuning approaches, limiting the contribution of question prompting.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2024. Llms' reading comprehension is affected by parametric knowledge and struggles with hypothetical statements. *arXiv preprint arXiv:2404.06283*.

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

William Cain. 2024. Prompting change: Exploring prompt engineering in large language model ai and its potential to transform education. *TechTrends*, 68(1):47–57.

Stefano Campese, Ivano Lauriola, and Alessandro Moschitti. 2023. QUADRo: Dataset and models for QUestion-answer database retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15573–15587, Singapore. Association for Computational Linguistics.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2024a. Unleashing the potential of prompt engineering: a comprehensive review. *Preprint*, arXiv:2310.14735.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Luca Di Liello, Siddhant Garg, and Alessandro Moschitti. 2023. Context-aware transformer pre-training for answer sentence selection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–468, Toronto, Canada. Association for Computational Linguistics.

Orna Raz Eitan Farchi Ateret Anaby-Tavor Ella Rabinovich, Samuel Ackerman. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the 3rd Version of the Generation, Evaluation Metrics (GEM) Workshop of The 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Matteo Gabburo, Rik Koncel-Kedziorski, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Knowledge transfer from answer ranking to answer generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9481–9495, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: A review-based question answering task. *arXiv preprint*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Zhibo Hu, Chen Wang, Yanfeng Shu, Helen, Paik, and Liming Zhu. 2024. Prompt perturbation in retrieval-augmented generation based large language models. *Preprint*, arXiv:2402.07179.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *CoRR*, abs/2403.02901.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2024. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint*.

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.

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. 2021a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021b. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18608–18616.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus. *Preprint*, arXiv:1911.02782.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA. Association for Computing Machinery.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *Preprint*, arXiv:2402.13228.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Teven Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth? *arXiv preprint arXiv:2103.08493*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

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.

Shubham Ugare, Tarun Suresh, Hangoo Kang, Sasa Misailovic, and Gagandeep Singh. 2024. Improving llm code generation with grammar augmentation. *arXiv preprint arXiv:2403.01632*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2021. Universal adversarial triggers for attacking and analyzing nlp. *Preprint*, arXiv:1908.07125.

Zizhen Wang, Yixing Fan, Jiafeng Guo, Liu Yang, Ruqing Zhang, Yanyan Lan, Xueqi Cheng, Hui Jiang, and Xiaozhao Wang. 2020. Match^2: A matching over matching model for similar question identification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 559–568, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

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.

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.

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.

Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. 2024. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

A List of prompts used

Simple QA prompt with no external information (no RAG, no SQs) Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should answer the question without external context.
If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding further context.
QUESTION: {question}
[/INST]

QA prompt that uses retrieved paragraphs Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should answer the question based on the provided context.
The context consists of N documents that are relevant to the input question.
If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding further context.
Here is context to help:
{paragraphs}
QUESTION: {question}
[/INST]

QA prompt that uses SQs (retrieved or generated) Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ) system able to answer questions.
You should answer the question based on the provided context.
The context consists of N questions similar and related to the input question (frequently asked) and helps you to reason and formulate the correct answer to the input question.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous questions.
+ Provide concise answers.
Here is context to help:
{context}
QUESTION: {question}
[/INST]

QA prompt that uses retrieved SQs and their answers Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ) system able to answer questions.
You should answer the question based on the provided context.
The context consists of N question-answer pairs, where the questions are similar and related to the input question. The answer from each pair is the correct answer for that particular question.
The question-answer pairs help you to reason and formulate the correct answer to the input question.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous questions.
+ Provide concise answers.
Here is context to help:
{context}

QUESTION: {question}
[/INST]

QA prompt that used PopQA-TP dataset (no RAG, no SQs) Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction: You are a powerful Question Answering System. You should answer the question without external context.
The answer must always be an entity or a list of entities separated by a comma.
The input question is about the "prop" topic.
If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding further context.
QUESTION: {question}
[/INST]

QA prompt that used for SQs with PopQA-TP dataset Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction:
You are an AI Assistant acting as a Frequently Asked Questions (FAQ) system able to answer questions.
You should answer the question based on the provided context.
The context consists of N questions similar and related to the input question (frequently asked) and helps you to reason and formulate the correct answer to the input question.
The input question is about the "prop" topic.
You must respect the following rules:
[RULES]
+ If you don't know the answer, don't try to answer; just say "I don't know" and avoid adding unnecessary information.
+ Do not explicitly state that you are a FAQ system.
+ Do not explicitly cite the documents you use to answer the question.
+ Do not mention what other people ask.
+ Use the context to infer missing information or to clarify ambiguous questions.
+ The answer must always be an entity or a list of entities separated by comma
Here is context to help:
{context}
QUESTION: {question}
[/INST]

Question generation Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction:
You are a powerful AI that given an input question generates 5 similar questions.
A similar question is a question that is asking for the same thing as the input but posed in a different manner or using different words or in a way that is not trivial for a language model.
You should generate 5 similar questions.
Rules:
+ Your output must be a valid JSON, just a JSON, not other text or information is allowed.
+ The structure of the JSON must follow this:
{ "q1": "generated question 1",
"q2": "generated question 2",
"q3": "generated question 3",
"q4": "generated question 4",
"q5": "generated question 5" }
+ The questions must be different from each other and from the input but express the same meaning and ask for the same thing.
+ The questions must require the same answer and the same documents to be answered.
+ Be sure that the output is valid JSON, escape where necessary.
+ If the definition or the meaning of a word/thing is asked in the input question, be sure the generated questions ask for the same word/thing meaning.
Here is a couple of examples to help:
Example 1:
input question: Can lizards fly?
generated questions:
{ "q1": "Can lizards fly through the air?",
"q2": "Do lizards fly?",
"q3": "Are there lizards which can fly?",
"q4": "Are there any flying reptiles?",
"q5": "Are there any flying lizards?" }

...other 3 examples are showed in the prompt...

input question: question [/INST] generated questions:

Chain-of-Thoughts Below is an instruction that describes a task. Write a response that appropriately completes the request.
[INST]
Instruction: Answer the question following the

reasoning process in Example 1 and Example 2.
If you don't know the answer, don't try to answer; just say "I don't know."
The output must be only the answer.

Example 1:
input question: At what temperature is a chicken done?
similar questions are:
+ What temperature does a chicken have to be done?
+ What is the temperature supposed to be in the chicken to be done?
+ What temperature should a whole chicken be cooked at?
+ What is the "internal temperature" of done chicken?
+ What temperature do you cook the chicken to?
if these are similar questions, then the answer is: All poultry should reach a safe minimum internal temperature of 165 °F (73.9 °C) as measured with a food thermometer.

Example 2:
input question: elegxo meaning
similar questions are:
+ What does the term elegxo signify?
+ Can you explain the meaning of elegxo?
+ What is the definition of elegxo?
+ What does elegxo mean in Greek?
+ How is elegxo used in a sentence and what does it mean?
if these are similar questions, then the answer is: The Ancient Greek term "elegxo" means to refute, expose, convict, or examine.

Let's begin:
input question: question
[/INST]

B QRS training

Starting from a public checkpoint of MiniLM-v2-12L, 33M parameters, we continuously pre-trained it on a plethora of datasets for unsupervised Sentence Text Similarity (STS) tasks, including paraphrasing, sentence similarity, question answering, and summarization to name a few. Some of these datasets are MSMARCO (Nguyen et al., 2016), Natural Questions (Kwiatkowski et al., 2019), The Semantic Scholar Open Research Corpus (Lo et al., 2020), PAQ (Lewis et al., 2021b), AmazonQA (Gupta et al., 2019), WikiHow (Koupaee and Wang, 2018), and many others. A comprehensive list can be found on the web[4]. Overall, these resources contain more than ≈ 0.9B semantically related text pairs.

Similarly to previous work on dense retrieval training, e.g. SentenceBERT (Reimers and Gurevych, 2019), we consider a simple pre-training task where the model predicts if two texts are semantically equivalent or not. The model was pre-trained with mixed precision (FP16), Symmetric MultipleNegativesRanking loss (Henderson et al., 2017), learning rate of 2e-5, batch size of 1536, and max sequence length of 128 tokens.

After pre-training, the model is fine-tuned on a labeled Question Ranking dataset (Campese et al., 2023). Our best configuration, selected through model selection, is based on MultipleNegatives-Ranking and Online Contrastive losses, learning rate of 5e-6 and batch size of 32.

C Annotation guidelines

We used Amazon Mechanical Turk (AMT) to annotate and verify answers from every LLM configurations we used in this paper. For each question answer pair, we asked annotators: *Given a question and its associate answer generated by an AI model (LLM), tell us if the answer is:*

Correct and natural The answer is correct and the text looks natural.
Example: [Q] *How many calories in a cucumber?* [A] *Each one-cup (104-gram) serving contains just 16 calories, while an entire 11-ounce (300-gram) cucumber contains only 45 calories.*
The information provided in the answer is correct as it contains useful information to fill the input information-seeking intent. Moreover, the answer looks very natural and precise. It does not contain repetitions or irrelevant text.

Correct but not natural The answer contains the information asked in the question, but it is not perfect. For instance, the answer may contain extra text which is irrelevant.
Example: [Q] *How many calories in a cucumber?* [A] *An average pineapple (900 gr) contains 452 calories, which is higher compared to the 45 calories of a whole cucumber.*

[4] https://www.sbert.net

The answer is correct but it provides additional non-requested information. Apparently, it is answering a different question but it still contain the information requested. For this reason, we consider this type of answer correct but not natural.

Incorrect but very natural (hallucination) The answer is not correct but it is coherent and looks correct.
Example: [Q]*How many calories in a cucumber?* [A] *A cucumber contains 150 calories per single ounce of product*.
The answer is not correct as it provides wrong information. However, it was generated in a way that looked natural and trustful. We believe that this type of error is even more dangerous than providing simply incorrect answers as the user may not notice the issue. We asked annotators to use Google to verify the provided information when their knowledge and background were not sufficient.

Incorrect The answer is not correct as it is not related at all to the question or the information provided is wrong.
Example: [Q] *How many calories in a cucumber?* [A] *A Banana is a low-calories fruit*.
The answer does not contain the requested information. Note that this differs from Incorrect-but-natural as the provided text is obviously wrong for a given user. Although there may be some subjectivity between the two incorrect labels, we treat both cases as negative output.

Each elementary annotation task, or Human Intelligent Task (HIT) consists of evaluating 6 questions and their answers. The 6 pairs selected for a single HIT include 1 control question and 5 actual question/answer pairs from our models to be evaluated. The selection of the HITs was random, meaning that a single HIT may contain different questions and answers from different models and configurations.

Control questions were designed to be simple enough to simply filter out distracted annotators. Some examples of control questions and their answers are:

We adopted the following criteria to select annotators: (i) HIT approval rate to be greater then 96%; (ii) minimum history of 1000 HITs previously approved for other tasks; (iii) master turkers only. Each HIT was paid 0.50$.

Positive Control Questions
Q: Who was the first person to step on the moon? **A:** Neil Armstrong
Q: Who wrote 'Romeo and Juliet'? **A:** William Shakespeare
Negative Control Questions
Q: What is the capital of France? **A:** Berlin is the capital city of France.
Q: What is the chemical symbol for helium? **A:** Helium's chemical symbol is H

Table 5: Examples of control questions

D Failure cases with Smaug

Table 6 shows some examples of questions from QR where Smaug fails to answer when using question prompt.

Q: Abraham was born in what year? **Answer base:** Abraham Lincoln was born in 1809. **Answer prompt:** I'm sorry, but as an AI text-based model, I don't have real-time information or access to specific personal data like birth dates of individuals. Abraham could refer to various people throughout history with different birth years
Q: When will the humans be probably extinct? **Answer base:** It's difficult to predict exactly when humans might become extinct, as it depends on various factors such as technological advancements, environmental changes, and global crises. Some experts believe our species could face significant challenges in the future, but it's impossible to determine a specific timeline for potential extinction. **Answer prompt:** I dont know
Q: who best soccer player in the world? **Answer base:** Lionel Messi and Cristiano Ronaldo are often considered as two of the best soccer players in the world due to their numerous achievements and individual awards throughout their careers. However, opinions may vary depending on personal preferences and specific time periods being evaluated. **Answer prompt:** I dont know

Table 6: Examples Smaug failures.