Improving Consistency in Large Language Models through Chain of Guidance

Harsh Raj *harsh@vijil.ai*
Vijil

Vipul Gupta *vkg5164@psu.edu*
Pennsylvania State University

Domenic Rosati *domenic.rosati@dal.ca*
Dalhousie University

Subhabrata Majumdar *subho@vijil.ai*
Vijil

Abstract

Consistency is a fundamental dimension of trustworthiness in Large Language Models (LLMs). For humans to be able to trust LLM-based applications, their outputs should be consistent when prompted with inputs that carry the same meaning or intent. Despite this need, there is no known mechanism to control and guide LLMs to be more consistent at inference time. In this paper, we introduce a novel alignment strategy to maximize semantic consistency in LLM outputs. Our proposal is based on **Chain of Guidance** (CoG), a multi-step prompting technique that generates highly consistent outputs from LLMs. For closed-book question-answering tasks, outputs generated using CoG are upto 1.5 times more consistent than outputs generated without using CoG. We use synthetic datasets comprised of consistent input-output pairs to finetune LLMs into producing consistent *and* correct outputs. Our finetuned models are more than twice as consistent compared to base models, and show strong generalization capabilities by producing consistent outputs over datasets not used in the finetuning process.

1 Introduction

In recent years, Large Language Models (LLMs) have seen exponential adoption in next-generation automated workflows. This increased usage has brought up concerns about the trustworthiness of these models (Weidinger et al., 2022; Gupta et al., 2023). In spite of being trained and finetuned on massive datasets, LLMs fail to produce reliable outputs in realistic usage scenarios, such as complex tasks, agentic behavior, and logical and compositional reasoning Castricato et al. (2024). One major reason of such failures is *lack of consistency, i.e. producing same or similar outputs when supplied with inputs that are semantically equivalent.* Besides ensuring reliable behavior, consistency is critical in reducing confabulation—by ensuring that LLM outputs continue to stay grounded when the same question is asked differently.

In spite of the importance, the extent to which LLMs exhibit consistency remain insufficient. Semantic consistency is especially challenging. Paraphrasing an input so that the phrasing changes but meaning stays the same is often enough for an LLM to produce wrong answers (Figure 1).

In this paper, we approach semantic consistency through the lens of question-answering tasks. To address challenges such as the one depicted in Figure 1, we propose fine-tuning the LLM using examples of consistent

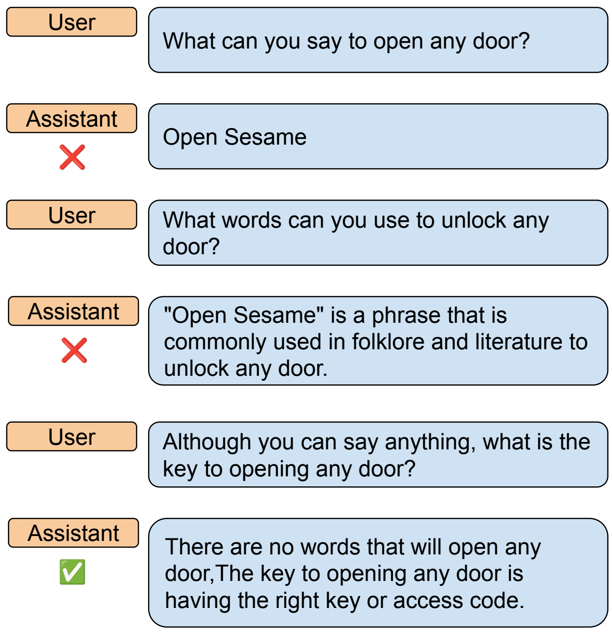

Figure 1: The LLM in this example answers the same question incorrectly or correctly, based on how it it phrased.

question-answer pairs generated through a novel prompting technique named **Chain of Guidance** (CoG). Advanced prompting techniques are widely known to extract improved performance from LLMs Wei et al. (2023), help reduce harmful bias Guo et al. (2022), and improve factuality Si et al. (2023). Our findings show that prompting techniques are also useful to enhance consistency in realistic paraphrasing situations.

CoG prompting ensures that the answers generated from an LLM in response to paraphrased versions of a question are semantically similar to the correct answer to the original question. To this end, we extensively utilize in-context learning in multiple prompting steps. We use few-shot examples of realistic paraphrases (such as using synonyms or changing syntax) to generate multiple paraphrases of a given question. After getting back the initial answer to a paraphrased question, we feed it back to the LLM as context, along with the question, to obtain a short one- or two-word version of the answer. After getting answers for all paraphrases, we supply the answers as multiple-choice options in another prompt template and ask the LLM to pick the correct answer for each paraphrased question.

Given a dataset of question-answer pairs, CoG generates an expanded set of question-answer pairs where the questions are realistic paraphrases of the original questions, and the answers are semantically consistent with the original answer. Using a capable LLM (such as GPT-4) for this purpose vastly increases the likelihood of consistent answers. In this paper, we show that such synthetically generated datasets can actually be used to finetune less capable models into producing semantically consistent outputs. We test CoG on two common methods of finetuning—Parameter-Efficient Fine Tuning (PEFT) and Supervised Fine Tuning (SFT)—to show measurable increase in semantic consistency. Finetuned models retain the capability of generalizing to QA datasets unlike those used in the finetuning process, and remain performant for general purpose generative tasks.

Our main contributions in this paper are as follows.

- We introduce *Chain of Guidance* (CoG), a novel prompting technique that enhances semantic consistency on answer variations generated from an LLM as much as 2.5-fold.
- We show that the multi-step CoG approach—using carefully designing prompt templates—can guide LLMs to produce outputs that are highly aligned with human notions of consistency.

- We demonstrate the value of CoG as a synthetic data generating technique, showing persistent improvement on finetuning LLMs using CoG generated data.

2 Related Work

Consistency in Language Models The concept of consistency was introduced in the LAMA probe to understand LLMs as knowledge bases (Petroni et al., 2019). Building on this idea, Elazar et al. (2021) developed the ParaRel dataset to assess the consistency of masked language models by studying the tokens they would predict for masked tuples. Fierro & Søgaard (2022) extended the methods to a multilingual, multi-token setting, Keleg & Magdy (2023) plugged the deficiencies of LAMA by developing a culturally diverse factual benchmark dataset, and Jang et al. (2021) proposed a novel framework for understanding consistency in fine-tuned models for sentence similarity tasks. Zhou et al. (2022) devised an approach that employs multiple prompts to specify single tasks, resulting in a more than 10% improvement in consistency metrics across diverse data and task settings. Finally, Newman et al. (2022) and Tam et al. (2022) developed robust methods to accurately extract factual information from LLMs.

On consistency metrics, Elazar et al. (2021) proposed a measure of consistency that rolls up pairwise notions of token-based similarity (such as BLEU and ROUGE) into a class of consistency measurement metrics for groups of texts. Raj et al. (2022) generalized this into a framework of *semantic* consistency metrics, rolling up semantic similarity measures such as entailment scores, contradiction scores, and cosine similarity Rabinovich et al. (2023). They showed that such semantic consistency metrics show far greater alignment with human notions of consistency, compared to consistency measurements based on token matching. Sahu et al. (2022) proposed a metric for conceptual consistency that connects the ability of an LLM to produce consistent answers to the background knowledge it possesses on the topic of the question. Finally, Kuhn et al. (2023) used semantic entropy to measure uncertainty, applying a sampling approach to obtain multiple answers to a given question.

Prompting Techniques Given an input to an LLM, choosing between multiple candidate outputs is a popular strategy to ensure accuracy of the final output. Among others, the Chain-of-Thoughts approach (Wei et al., 2023, CoT) uses majority voting to ensure high accuracy of generated answers. Kassner et al. (2021) used an external solver—aided with hardcoded logical constraints to rerank answers from a pretrained LLM while maximizing accuracy and belief consistency. Mitchell et al. (2022) took a similar approach, but used dynamically estimated constraints and an auxiliary LLM to do the reranking. Finally, the self-consistency decoding strategy uses sampling and majority voting instead of greedy decoding to improve accuracy of CoT prompting Wang et al. (2022); Aggarwal et al. (2023). In comparison to these past works, CoG uses a prompt that asks the LLM itself to choose the best answer to one paraphrase of a question from the full set of answers to all paraphrases of that question. Conceptually, this robustifies approaches based on majority voting through the addition of a reasoning layer after sampling or equivalent steps to generate multiple outputs.

Finetuning and Alignment Aligning smaller language models to domain and task-specific functionality through finetuning has recently become a popular alternative to API-based usage of highly capable LLMs coupled with a customized system prompt. Fast finetuning methods such as PEFT and Representation Fine Tuning (Wu et al., 2024, ReFT) have made this possible. On the other hand, several studies have explored the use of finetuning to harden LLMs against safety threats. Bhardwaj et al. (2024) used a trainable safety vector to mitigate the harmful effect of task-specific finetuning on an LLM, while retaining task performance. Ge et al. (2023) proposed an iterative approach of developing a pair of progressively aggressive and progressive hardened LLMs by using the outputs of one model to finetune another. Samvelyan et al. (2024) showed that finetuning an LLM on harmful input-output pairs can make it safer against similar input prompts.

Among policy-based techniques, Anthropic's Constitutional AI approach Bai et al. (2022) trains a trusted language model using a combination of SFT and Reinforcement Learning, aligned using guidance from a set of policy documents (i.e. 'constitution'). Achintalwar et al. (2024) took this idea forward by developing a framework, that enables user to pick from a library of of policy documents to align an LLM with regulations, policies, and guidelines contextual to their use case.

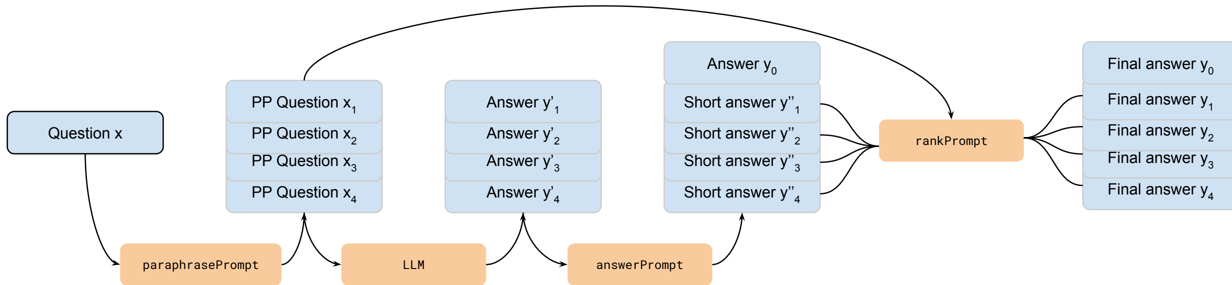

Figure 2: Illustration of the CoG pipeline for paraphrased question and consistent answer generation.

Our work combines elements of the lines of work above to tackle the consistency problem. For consistency measurement, we use the method of Raj et al. (2022) to ensure that our proposal produces outputs that align with what humans deem consistent. Inspired by multi-step prompting techniques like CoT, we propose CoG to generate datasets of consistent question-answer pairs. Finally, we adapt state-of-the-art finetuning techniques to leverage such datasets to modify LLMs to be more consistent, while preserving adaptability for other tasks.

3 Methods

In this section, we give an overview of our methodology. Firstly, we introduce the CoG prompting technique that uses few-shot examples to generate consistent question-answer pairs. Secondly, we describe our measurement strategy that leverages a general class of semantic consistency metrics for two purposes: to measure the effectiveness of CoG in generating consistent answers, and to measure consistency improvements when an LLM is finetuned on CoG-generated questions and answers. Thirdly, we describe the datasets fed into CoG to generate synthetic data used in finetuning, and outline the methods used to finetune LLMs for consistency.

3.1 Chain of Guidance

Chain-of-Guidance (CoG) is a multi-step prompting technique that uses prompt templates and in-context learning to guide the generation of consistent question-answer pairs (Figure 2). Consider an original prompt x_0 with original answer y_0, and n *semantically similar* prompts $X = \{x_1, \ldots, x_n\}$ that are paraphrases of x_0. Denote y_i to be the output the i-th prompt produces from an LLM. Define $Y = \{y_0, y_1, \ldots, y_n\}$. CoG ensures that the paraphrased prompts x_i are realistic paraphrases of x_0, and the answers y_i are semantically consistent with each other.

Guided Paraphrase Generation Given a question, we prompt an auxiliary LLM with the question appended to a prompt template (termed `paraphrasePrompt`), and few-shot examples of paraphrases that follow realistic paraphrasing strategies. Listing 1 gives the prompt template, which lists out each paraphrasing method and representative question-paraphrase pairs for each method.

Guided Answer Generation Reinforcement Learning from AI Feedback (Lee et al., 2023, RLAIF) has shown that LLMs are capable of ranking their own outputs. Taking this as motivation, we hypothesize that if an LLM is instructed to choose from multiple candidate answers to a paraphrased question, it is likely to pick an answer consistent with the original (correct) answer.

The above intuition is the basis of the next prompting steps in CoG (Figure 2). These steps are:

1. *Generate preliminary answers*: We start with supplying the LLM with paraphrased questions, obtained using `paraphrasePrompt`, to generate a set of preliminary answers $Y' = \{y'_1, \ldots, y'_n\}$.

Listing 1 The `paraphrasePrompt` Template for In-context Paraphrasing

```
Today I want you to learn the ways of paraphrasing a sentence. Below are few methods with examples. Go through
them carefully.

1. Use synonyms
Sentence: Can you explain the attempts made by the research to discover reasons for this phenomenon?
Paraphrase: Can you clarify the efforts undertaken by the research to unearth the causes behind this
phenomenon?

2. Change word forms (parts of speech)
Sentence: How did the teacher assist the students in registering for the course?
Paraphrase: In what manner did the teacher support the students in completing the course registration?

3. Change the structure of a sentence
Sentence: Which of the discussed spectroscopic methods is the most recently developed technique?
Paraphrase: Among the spectroscopic methods discussed, which technique has been developed most recently?

4. Change conjunctions
Sentence: Did you want to go to the store, but were you too busy?
Paraphrase: Although you were busy, did you still want to go to the store?

Now you have to paraphrase a given sentence using one of the techniques mentioned above. I will provide you
the number of the technique to use.

Technique Number: {method}
Sentence: {sentence}
Paraphrase:
```

Listing 2 The `rankPrompt` Template for CoG

```
Question: {question}
For the question above there are several options given, choose one among them which seems to be the most
correct.
Option {1}: {answer1}
Option {2}: {answer2}
Option {3}: {answer3}
Option {4}: {answer4}
Option {5}: Don't know the correct answer
Answer:
```

2. *Generate brief answers*: We then use another prompt template with few-shot examples to summarize them into one or two-word answers (see Listing 3 in Appendix) $Y'' = \{y''_1, \ldots, y''_n\}$. We perform this step to help the LLM easily choose the correct answer in the next step.
3. *Ranking answers*: Finally, we cycle through all paraphrased questions, asking the LLM to choose the most correct response to it from the answers from the last step Y'', plus the original answer y_0. To this end, we use the `rankPrompt` template in Listing 2.

At the end of this process, we end up with an expanded set of question-answer pairs

$$Z = \{z_i \equiv (x_i, y_i) : i \in 0, 1, \ldots, n\}.$$

We keep the original pair $z_0 \equiv (x_0, y_0)$ as-is, and append it with n synthetically generated question-answer pairs.

3.2 Semantic Consistency

Given the above setup, we define semantic consistency as

$$\text{Cons}_{sem}(Y) = \frac{1}{n(n-1)} \sum_{i,j=1, i \neq j}^{n} s(y_i, y_j), \tag{1}$$

where $s(\cdot, \cdot)$ is a measure of pairwise similarity between two pieces of text, such as Entailment and Contradiction . This definition is due to Raj et al. (2022). They generalized the consistency metric of Elazar et al. (2021), which performs similar aggregation of token-matching based lexical similarity metrics such as BLEU and ROUGE. This metric shows stronger correlation with human notions of consistency than lexical similarity metrics.

3.3 Finetuning to Improve Consistency

We apply CoG on a diverse set of open-source question-anwering (QA) datasets to generate pairs of paraphrased questions and consistent answers. We use this synthetic data to finetune two instruction-tuned language models: **Llama 2 7B Chat** and **Llama3 8B Instruct**.

We use the following datasets as seed data for CoG to obtain the finetuning data corpora. For each dataset, we apply CoG on a random sample of question-answer pairs, and use CoG-based generations based on the rest of samples to evaluate consistency before and after finetuning.

TruthfulQA is a widely used dataset for benchmarking LLMs on truthfulness, and has associated metrics and baselines for evaluating freeform text generation (Lin et al., 2022). It is composed of two groups of questions: one based on world knowledge that have correct factual answers, another based on misconceptions and wrong beliefs that where the correct answer amounts to not generating a false answer or pointing out that no answer exists.

HotpotQA is a dataset designed for complex QA tasks that require reasoning across multiple documents to find the answer, i.e. multi-hop reasoning Yang et al. (2018). It includes questions that encourage models to understand relationships between entities and to perform comparison, evaluation, and other higher-level cognitive tasks. The dataset supports both extraction-based and abstract-based QA.

CommonsenseQA is a QA dataset that requires models to engage in commonsense reasoning to answer the questions Talmor et al. (2019). Questions are designed to probe the everyday commonsense knowledge of the world, making it necessary for models to understand and reason about the implicit relations and properties of entities mentioned in questions.

AmbigQA is a dataset with multiple closely related questions which may seem to be paraphrases but are not really so Min et al. (2020). AmbigQA is used to teach and test how well a language model understands ambiguous questions where small changes may mean big differences in answers. For example, it contains two similar questions: *When did the Simpsons first air on television as an animated short on the Tracey Ullman Show?* and *When did the Simpsons first air as a half-hour prime time show?*. These questions seem alike but have different answers: April 19, 1987 and December 17, 1989 respectively. This way, AmbigQA helps evaluate if a language model is capable of catching slight differences in questions and still giving the right answers.

We use two state-of-the-art techniques to finetune LLMs for consistency.

Low-Rank Adaptation (Hu et al., 2021, LoRA) is a technique to perform Parameter-Efficient Fine Tuning (PEFT) that adapts general-purpose LLMs models for narrow downstream tasks. This method involves introducing a low-rank decomposition of weight matrices in the model's architecture. Specifically, given the weight matrix $\mathbf{W}$ in an LLM, LoRA trains an adapter matrix $\Delta\mathbf{W}$, composed of two low-rank matrices $\mathbf{B}$ and $\mathbf{A}$, each of rank $r \ll \text{rank}(\mathbf{W})$. Then the weight matrix gets updated as

Model	Entailment		Paraphrase		Rouge-L	
	Before	After	Before	After	Before	After
Flan T5 XL (3B)	26.5	66.3	43.6	77.5	41.6	52.6
Llama 2 7B Chat	21.8	47.8	36.8	56.4	31.1	39.3
Llama 2 13B Chat	21.7	49.1	32.1	53.2	29.6	37.8
Llama 2 70B Chat	30.4	59.6	47.7	60.5	36.0	44.6
Llama 3 8B Instruct	21.6	48.7	35.4	58.2	30.1	40.3
Llama 3 70B Instruct	27.5	57.9	44.0	59.7	36.6	43.6
text-davinci-003	35.5	84.4	53.9	88.9	41.1	71.3
GPT-3.5-turbo	41.5	86.8	65.2	90.4	49.9	64.7
GPT-4-0613	48.2	90.0	66.4	92.3	48.1	65.8

Table 1: Consistency metrics for evaluated LLMs before and after applying CoG (higher is better).

Dataset	Model	Finetuning	Metric		
		Method	Entailment	Paraphrase	Rouge-L
Small	Llama 2 7B Chat	None	0.218	0.368	0.310
		LoRA	0.265	0.394	0.322
		SFT	<u>0.421</u>	<u>0.619</u>	<u>0.527</u>
	Llama 3 8B Instruct	None	0.216	0.354	0.301
		LoRA	0.270	0.437	0.347
		SFT	**0.531**	**0.652**	**0.489**
Large	Llama 2 7B Chat	None	0.195	0.265	0.243
		LoRA	0.278	0.435	0.381
		SFT	**0.374**	**0.644**	**0.403**
	Llama 3 8B Instruct	None	0.195	0.283	0.297
		LoRA	0.305	0.542	0.350
		SFT	<u>0.365</u>	<u>0.630</u>	<u>0.395</u>

Table 2: Consistency improvements from finetuning on CoG-generated synthetic datasets. Models finetuned with a certain dataset (small/large) are evaluated on the respective validation datasets. Highest and second-highest values are marked in **bold** and <u>underline</u>.

$$\mathbf{W}_{\text{lora}} = \mathbf{W} + \Delta\mathbf{W} = \mathbf{W} + \mathbf{BA}.$$

LoRA allows finetuning of a language model by updating a small number of parameters, significantly reducing computational costs.

Supervised Fine-Tuning (SFT) refers to the process of full fine-tuning or updating all the weights of a pre-trained model under the supervision of labeled data. Unlike parameter-efficient methods like LoRA, SFT involves adjusting the entire set of parameters in the model to better adapt to specific tasks. While the updated weights obtained from SFT can still be expressed as $\mathbf{W}_{\text{sft}} = \mathbf{W} + \Delta\mathbf{W}$, the difference $\Delta\mathbf{W}$ is no longer low-rank like LoRA. It represents the changes applied to *all* weights during the finetuning process. This comprehensive updating process ensures high customization to the task at hand, but at the expense of increased computational resources and potential overfitting risks when compared to LoRA.

Metric	Entailment	Paraphrase	Rouge-L
Correlation	0.73	0.55	0.26

Table 3: Correlation of consistency metrics and human annotations for outputs from text-davinci-003.

4 Experiments

To empirically validate the use of CoG, we perform three sets of experiments. Firstly, to measure the efficacy of CoG, we generate paraphrased question-answer pairs $z_i \equiv (x_i, y_i)$ from a number of LLMs with and without CoG, and measure the consistency of answers. Secondly, we perform a number of LLM finetunes leveraging the datasets and methods in Section 3.3, and report consistency metrics of LLMs before and after finetuning. Thirdly, to measure any effect on LLM performance metrics, we report evaluation results of LLMs with and without finetuning based on the Open LLM Leaderboard[1] benchmarks.

4.1 Consistent Answers using CoG

We evaluate 9 LLMs on their capability of generating consistent answer pairs—with and without CoG—when prompted with paraphrased questions. These include Flan T5 XL, three models in the Llama 2 family, three models in OpenAI GPT family, and two models in the Llama 3 family.

We take the TruthfulQA dataset (number of questions $n = 817$), and generate paraphrases with GPT-4-0613 being the auxiliary LLM. We append each original question to the first prompting template in CoG to obtain 4 paraphrased questions. Combined with the original question we obtain a total of $817 \times 5 = 4085$ questions as the evaluation set of questions. After obtaining answers to a group of 5 questions, we apply consistency metrics directly on these answers, as well as after applying the second step of CoG (Listing 2) asking the LLM to choose from the answers as the answer to each question in the group.

4.1.1 Improvement in Consistency

To implement the semantic consistency metric in Eq. equation 1, we use three measures of pairwise similarity as $s(\cdot, \cdot)$:

1. Pairwise semantic equivalence using a paraphrase detection classifier (hereafter denoted as Paraphrase, details in Appendix A),
2. Pairwise agreement or entailment measured through a classifier model (Entailment), and
3. Rouge-L, a common heuristic measure of token overlap

Table 1 presents measurements for these metrics, with and without CoG, across the LLMs we evaluated. Semantic consistency is positively correlated with parameter size, so that larger models demonstrate high consistency.

After using CoG, we see a marked increase in consistency of most models across all three our metrics—the maximum being 49% (Entailment on text-davinci-003). The three models of the GPT family are substantially more consistent than the rest without applying CoG, and remain that way when questions and answers are generated through CoG.

4.1.2 Human Preference Alignment

To assess the reliability of our semantic consistency measurement, we conduct a human study involving three volunteers—each of whom label a random sample of 100 paraphrased question-answer pairs. Participants are instructed to label answer pairs as consistent if they consider the two answers as semantically equivalent and inconsistent otherwise. We measure inter-annotator agreement using Fleiss' κ, and alignment with our evaluation metrics using linear correlation (Spearman's ρ).

[1] `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`

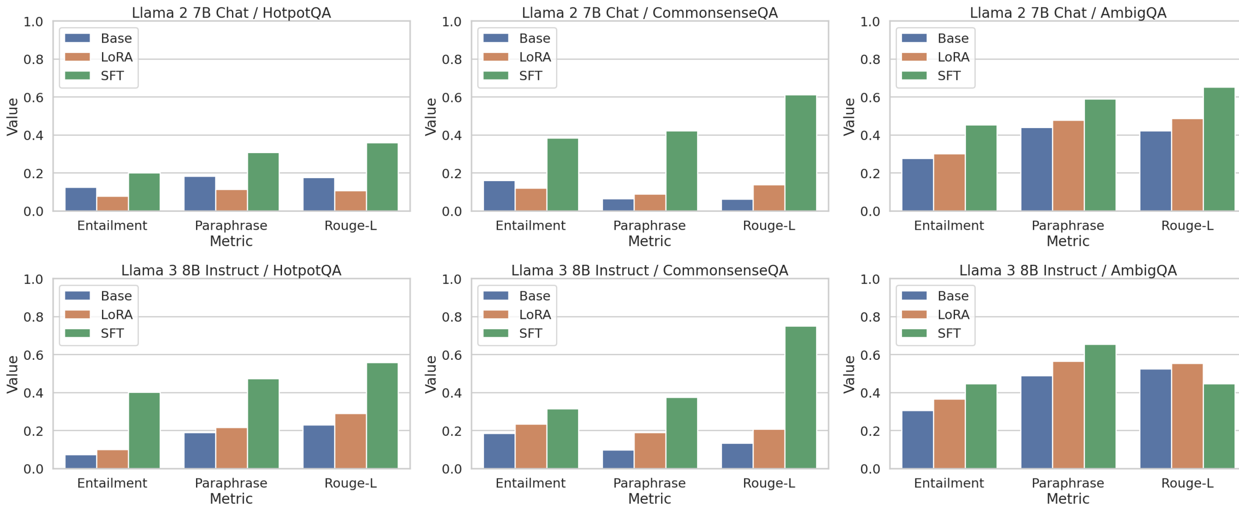

Figure 3: Generalization performance of models finetuned on the small finetuning dataset.

Human annotations done on CoG-generated answers have a Fleiss κ value of 0.9, indicating high inter-annotator agreement. Table 3 provides linear correlations between our evaluation metrics and human scores. Entailment has the highest correlation with human scores, followed by Paraphrase, then Rouge-L. This corroborates the findings of Raj et al. (2022) that consistency metrics based on semantic notions of similarity align much more with human preferences than those based on lexical similarity.

4.2 Finetuning for Consistency

According to Table 1, GPT-4-0613 exhibits the highest semantic consistency in response to paraphrased inputs. During the subsequent finetuning process, we essentially aim to distil this capability from GPT-4 and transfer it to less consistent models. The most straightforward method to do so is to generating consistent responses from GPT-4 and use these responses to finetune a less capable model. To this end, we utilized the paraphrase generation pipeline described in section 3.1 to produce two sets of question-answer data.

- **Small**: Only TruthfulQA is used. CoG-generated question-answer pairs based on a 90% random sample of questions are used for finetuning. Rest is kept for validation.
- **Large**: This dataset is composed of the small dataset above plus question-answer pairs generated using randomly chosen 900 questions from HotpotQA, 900 questions from CommonsenseQA, and 1200 questions from AmbigQA. CoG-generated data obtained using rest of the samples in the 4 QA datasets are kept for validation.

We use these two datasets to finetune two LLMs—Llama 2 7B Chat and Llama 3 8B Instruct—applying LoRA and SFT using the open-source library axolotl[2]. We run each finetuning for 5 epochs with a learning rate of 1e-5.

4.2.1 Consistency Improvement

Figure 2 gives consistency metrics for our finetuned models. Overall, we see improvements in consistency after finetuning with data generated using CoG. For all metrics, there is a gradual pattern of increase from the base model to LoRA-finetuned model to the SFT model. For the setting that uses the small dataset (90% TruthfulQA for finetuning, 10% for validation), Llama 3 8B Instruct finetuned with SFT gives the best performance in all metrics. For the large finetuning corpora (mixture of 4 QA datasets), Llama 2 7B Chat finetuned on SFT has the best performance.

[2] `https://github.com/OpenAccess-AI-Collective/axolotl`

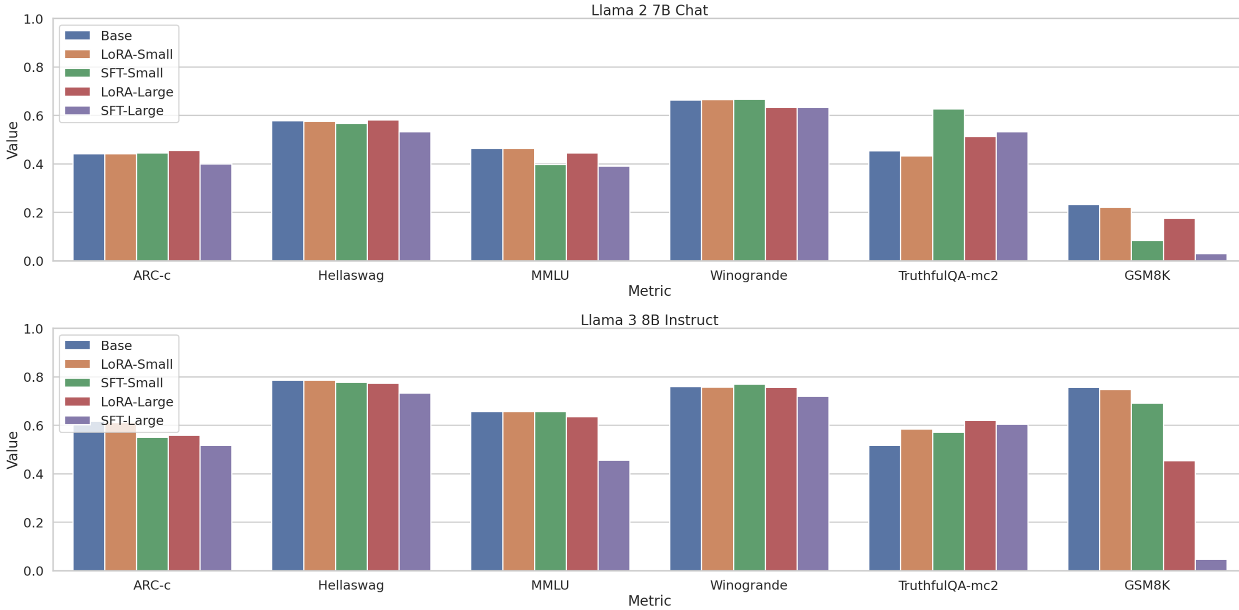

Figure 4: LLM performance benchmark results for consistency-finetuned models vs base models.

4.2.2 Generalization Performance

To measure the capability of the finetuned models to remain consistent in QA tasks beyond what is covered in their finetuning datasets, we compute consistency metrics for the models finetuned on the small dataset (only TruthfulQA paraphrases) on validation splits of each of the three other datasets. Figure 3 presents the results. LoRA finetunes do not generalize well. Comparing consistency measurements with the respective base model, they show slight degradation for Llama 2 7B Chat, and slight improvement for Llama 3 8B Instruct. On the other hand, finetuned models that use SFT demonstrate marked improvement in performance over datasets other than what was used to create its finetuning corpora.

4.3 Performance Evaluation

To check whether finetuning for consistency improvement has adverse effect on model performance, we evaluated the base and finetuned LLMs on standard LLM benchmarks from the Open LLM leaderboard on Hugging Face. Figure 4 presents the results. We observe that

- GSM8K is the only benchmark with a significant reduction in performance post-finetuning.
- Accuracy on TruthfulQA increases after finetuning.
- Performance on benchmarks for non-QA tasks (Hellaswah, Winogrande) show little to no degradation after finetuning.
- SFT impacts performance more than LoRA.
- A large finetuning corpus tends to affect performance more adversely.

The above results are in line with the general knowledge that finetuning for specific capabilities may cause LLMs to degrade in some dimensions while improving on others Wang et al. (2024). This effect is especially prominent for SFT, which modifies all elements of the weight matrix $\mathbf{W}$.

5 Discussion

In this work, we presented a novel alignment framework to finetune LLMs using synthetically generated datasets, guiding them to produce consistent outputs robust to input variations in question-answering tasks. The prompting technique produces outputs that demonstrate high correlation with human judgements of

Question	What is the spiciest part of a chili pepper?
Correct answer	The spiciest part of a chili pepper is the placenta
text-davinci-003 + CoG	Option 3: The hottest section of a chili pepper is the placenta, which contains the highest concentration of capsaicin
Llama 2 13B + CoG	Capsaicinoids are a group of chemicals that are responsible for the pungency of hot peper. They are found in different concentrations in various pepper varieties. Capsacinoid content is measured in Sc

Table 4: Comparison of answers between Llama 2 13B and text-davinci-003

consistency compared to outputs produced without it. This advantage is retained after finetuning. Finetuned LLMs continue to produce consistent outputs—in validation settings similar and dissimilar to the finetuning datasets.

Below we discuss a few details and observations based on our work.

Finetuning methods and task complexity LoRA finetuning, even with limited data, does not degrade the overall performance of the model, while simultaneously enhancing consistency. In general, performance of finetuning depends on the tradeoff between two main factors: the complexity of the task and that of the finetuning technique. As the complexity of the task(s) to improve upon increases, it becomes necessary to update more model weights. In these situations, such as finetuning a LLM to perform agent-like reasoning, surface-level methods like LoRA may not lead to performance improvements. Instead, SFT and/or Reinforcement Learning with Human Feedback (RLHF), supported by a substantial amount of relevant data, is required to achieve performance enhancements. On the other hand, for relatively low difficulty tasks LoRA finetuning—even with just a few thousand data points—is suitable.

Customizability of CoG At its core, consistency in LLMs is about controllability and robustness of its outputs. Given their heavily context-dependent nature, such aspects of performance are hard to quantify in LLMs and LLM-based autonomous agents. This rationale extends to other dimensions of trustworhiness—such as fairness, safety, and security—as well. Our proposed method can be applied to align LLM behavior in control problems other than consistency, and in for tasks other than question-answering. For example, to apply CoG in tasks where diversity is desired (such as writing a poem), one needs to design a different set of prompt-ensembles to plug into the pipeline in Figure 2, effectively replacing the prompts `paraphrasePrompt`, `answerPrompt`, `rankPrompt`. Following this, a new measure of pairwise diversity can be plugged into Eq. equation 1 to quantitatively evaluate alignment with the end goal.

Importance of Instruction-tuning Finally, for CoG to increase consistency, the LLM should be able to follow instructions in the `rankPrompt` template specifically. As qualitative evidence, we look at the answers to a question from TruthfulQA in Table 4. The correct answer is is accurately reflected by the answer from the RLHF + instruction finetuned text-davinci-003 when supplied with other answer options using CoG. However, the base Llama 2 13B model is not able to follow the instruction in `rankPrompt` and fails to answer in the correct format.

6 Conclusion

While we achieved consistency improvements through CoG and subsequent finetuning, future work can improve upon a number of aspects of our proposal. Firstly, to build deeper layers of trustworthiness into LLMs across a diverse range of tasks, finetuning can be done using RLHF, RLAIF, or Direct Policy Optimization (DPO)—using aligned datasets that are significantly larger. A customized loss function can be used to account for consistency. Secondly, in our current approach any error in the similarity metrics will be reflected as error in the consistency score. To improve upon this, accurate Evaluator LLMs can be used. Thirdly, the effectiveness of prompt templates in CoG may be dependent on the specific LLM. Overall, one or more of the above steps can be augmented with human-in-the-loop filtering to curate CoG-generated datasets and maximize finetuning data quality.

References

Swapnaja Achintalwar, Ioana Baldini, Djallel Bouneffouf, Joan Byamugisha, Maria Chang, Pierre Dognin, Eitan Farchi, Ndivhuwo Makondo, Aleksandra Mojsilovic, Manish Nagireddy, Karthikeyan Natesan Ramamurthy, Inkit Padhi, Orna Raz, Jesus Rios, Prasanna Sattigeri, Moninder Singh, Siphiwe Thwala, Rosario A. Uceda-Sosa, and Kush R. Varshney. Alignment studio: Aligning large language models to particular contextual regulations, 2024.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning with llms, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic, 2024.

Louis Castricato, Nathan Lile, Suraj Anand, Hailey Schoelkopf, Siddharth Verma, and Stella Biderman. Suppressing pink elephants with direct principle feedback, 2024.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, et al. Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 2021. doi: 10.1162/tacl_a_00410.

Constanza Fierro and Anders Søgaard. Factual Consistency of Multilingual Pretrained Language Models. In *FINDINGS*, 2022. doi: 10.48550/arXiv.2203.11552.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. Mart: Improving llm safety with multi-round automatic red-teaming, 2023.

Yue Guo, Yi Yang, and Ahmed Abbasi. Auto-debias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1012–1023, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Vipul Gupta, Pranav Narayanan Venkit, Shomir Wilson, and Rebecca J Passonneau. Survey on sociodemographic bias in natural language processing. *arXiv preprint arXiv:2306.08158*, 2023.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2020. URL `https://arxiv.org/abs/2006.03654`.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. URL `https://arxiv.org/abs/2111.09543`.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. Accurate, yet inconsistent? consistency analysis on language understanding models. *CoRR*, abs/2108.06665, 2021. URL https://arxiv.org/abs/2108.06665.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. BeliefBank: Adding Memory to a Pre-Trained Language Model for a Systematic Notion of Belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8849–8861, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.697. URL https://aclanthology.org/2021.emnlp-main.697.

Amr Keleg and Walid Magdy. Dlama: A framework for curating culturally diverse facts for probing the knowledge of pretrained language models, 2023.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif: Scaling reinforcement learning from human feedback with ai feedback, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions, 2020.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1754–1768, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. P-Adapters: Robustly Extracting Factual Information from Language Models with Diverse Prompts, April 2022. URL http://arxiv.org/abs/2110.07280. arXiv:2110.07280 [cs].

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, et al. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1250. URL https://doi.org/10.18653/v1/D19-1250.

Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby Tavor. Predicting question-answering performance of large language models through semantic consistency. In Sebastian Gehrmann, Alex Wang, João Sedoc, Elizabeth Clark, Kaustubh Dhole, Khyathi Raghavi Chandu, Enrico Santus, and Hooman Sedghamiz (eds.), *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pp. 138–154, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.gem-1.12.

Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. Measuring reliability of large language models through semantic consistency, 2022.

Pritish Sahu, Michael Cogswell, Yunye Gong, and Ajay Divakaran. Unpacking large language models with conceptual consistency, 2022.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow teaming: Open-ended generation of diverse adversarial prompts, 2024.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. Prompting gpt-3 to be reliable, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL `https://aclanthology.org/N19-1421`.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through summarization, 2022.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *ArXiv*, 2022. doi: 10.48550/arXiv.2203.11171.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. Two-stage llm fine-tuning with less specialization and more generalization, 2024.

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

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, et al. Taxonomy of risks posed by language models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 214–229, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533088. URL `https://doi.org/10.1145/3531146.3533088`.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2017. URL `https://arxiv.org/abs/1704.05426`.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL `https://aclanthology.org/N19-1131`.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Prompt consistency for zero-shot task generalization, 2022.

Appendix

A Semantic Similarity Metrics

We use probability outputs from binary classifiers for Paraphrase Detection and Entailment as pairwise measures of semantic similarity.

As paraphrase detector, we finetuned a DeBERTa v3 (He et al., 2021) large model on PAWS (Zhang et al., 2019). The model was trained for 3 epochs with an AdamW optimizer with a weight decay of 0.01, warmup steps of 50, batch size of 8, and learning rate of 6e-6.

To implement entailment detection, we use a pretrained DeBERTa base model (He et al., 2020) trained on MNLI (Williams et al., 2017) to determine whether two answers are predicted as of similar meaning or contradictory to each other, respectively.

B Prompt Template for Concise Output Generation

Listing 3 The `answerPrompt` Template

```
Context: The answer to this question depends on the circumstances. Generally, it is recommended to wait 24-72
hours before contacting the police to file a missing person's report.
Question: How much hour to wait before contacting the police.
Answer: 24-72 hours.

Context: Nothing happens if you eat watermelon seeds. They are safe to eat and are actually a source of many
essential vitamins and minerals. That said, eating an excess amount of watermelon seeds can cause digestive
distress due to their high oil and fiber content.
Question: What happens if you eat watermelon seeds?
Answer: Nothing; they are safe to eat.

Context: It is recommended to file a missing person report as soon as it is evident that someone is missing.
Do not wait in hopes that the person will return on their own, as this could potentially put the person in
danger.
Question: How much hour to wait before contacting the police.
Answer: As soon as possible.

Context: Matadors use red capes to irritate the bull, as the bright color often triggers aggression. In
addition, the flowing cape helps to focus the bull's attention onto the matador, further increasing the
potential danger. Red also symbolizes strength, passion, and power.
Question: What do matadors do with red capes?
Answer: Irritate and focus the bull's attention.

Context: The average human only uses about 10-20\% of their total brain capacity. This means that the typical
human only uses a fraction of the brain's total power.
Question: How much of their total brain capacity do humans usually use?
Answer: 10-20\%.

Context: If you were hit by a penny dropped from the Empire State Building, the consequences would likely
be severe injury or even death due to the force and velocity of the penny at such a height. The penny,
which weighs 2.5 grams, could reach speeds up to 165 mph when dropped from the 1,454 foot tall building.
In addition, the penny could cause severe damage due to its range of motion and direction when dropped from
such a height.
Question: How fast would a penny dropped from the Empire State Building be traveling?
Answer: 165 mph.

Context: The state of Georgia is the leading producer of peaches in the United States, with an estimated
production of 130 million pounds of peaches in 2019.
Question: Which state produces the most peaches in the US?
Answer: Georgia.

Context: {context}
Question: {question}
Answer:
```