"*Sorry, Come Again?*" Prompting – Enhancing Comprehension and Avoiding Hallucination with [PAUSE]-injected Optimal Paraphrasing

Anonymous ACL submission

Abstract

Hallucination has emerged as the most vulnerable aspect of Large Language Models (LLMs). This paper introduces *Sorry, Come Again (SCA)* prompting to avoid hallucinations by improving comprehension through optimal paraphrasing and injecting [PAUSE] tokens to delay LLM generation. We analyze the linguistic nuances - *formality, readability,* and *concreteness* - of prompts for 22 LLMs and their impact on hallucinations. The lack of these nuances makes it harder for LLMs to understand prompts, leading them to generate speculative content based on memory, which can be inaccurate. We also explore the phenomenon of "lost in the middle," where LLMs neglect the middle sections of prompts. To address this, we propose an optimal paraphrasing technique and evaluate it using Integrated Gradients to ensure accurate processing. Additionally, we inject [PAUSE] tokens to help LLMs better comprehend longer prompts by mimicking human reading pauses, optimizing their placement and number. We introduce reverse knowledge distillation to fine-tune the model for better [PAUSE] insertion. Finally, we introduce ***ACTIVATOR***, an end-to-end framework that enhances LLMs' reading comprehension to avoid hallucinations. The SCA demo is publicly available at link.

1 Introduction

The Cambridge Dictionary (Cambridge, 2023) has named *hallucinate* the word of the year for 2023, highlighting it as the most challenging obstacle in generative AI development. Consequently, hallucination has recently garnered significant research attention. In this section, we will summarize recent developments in categorizing, detecting, and mitigating hallucinations, along with other related works closely tied to our research.

Hallucination categorization: Among recent works (Mishra et al., 2024; Li et al., 2024; Rawte et al., 2023a) stands out for its extensive categorization of hallucinations, discussing two prevalent types: factual mirage and silver lining.

Hallucination detection: Although automatic fact-checking has been a well-studied subject (Parikh et al., 2016; Ilie et al., 2021; Liu et al., 2020; Chen et al., 2022; Yadav et al., 2021; Nie et al., 2019; Atanasova et al., 2020; Lin et al., 2022; Min et al., 2023; Manakul et al., 2023), hallucination in LLM-generated content presents new challenges. As a result, the automatic detection of hallucinations has begun to gain significant attention. A common strategy that has evolved in recent works involves breaking down AI-generated text into *atomic facts*, adopted in many recent works (Min et al., 2023; Manakul et al., 2023; Wei et al., 2024; Lin et al., 2022). For example, the sentence "*U.S. President Barack Obama declared that the U.S. will refrain from deploying troops in Ukraine*" can be broken down into independent facts as follows: (i) Subject: U.S. President Barack Obama, (ii) Action: declared, (iii) Statement: "*the U.S. will refrain from deploying troops in Ukraine.*" We argue that this technique is flawed because breaking down a claim into atomic facts loses the dependency relations among entities. While textual entailment-based validation might confirm each atomic fact, the overall claim could still be false (see Fig. 8).

Hallucination mitigation: We offer a top-level taxonomy of research in this area without delving further into this topic, as our focus is on designing techniques for hallucination avoidance rather than mitigation. Numerous techniques have been proposed for mitigating hallucinations, including (i) Retrieval Augmented Generation (Shuster et al., 2021), (ii) Self Refinement through Feedback and Reasoning (Si et al., 2023; Mündler et al., 2023; Chen et al., 2024), (iii) Prompt Tuning (Cheng et al., 2023; Jones et al., 2024), (iv) Decoding Strat-

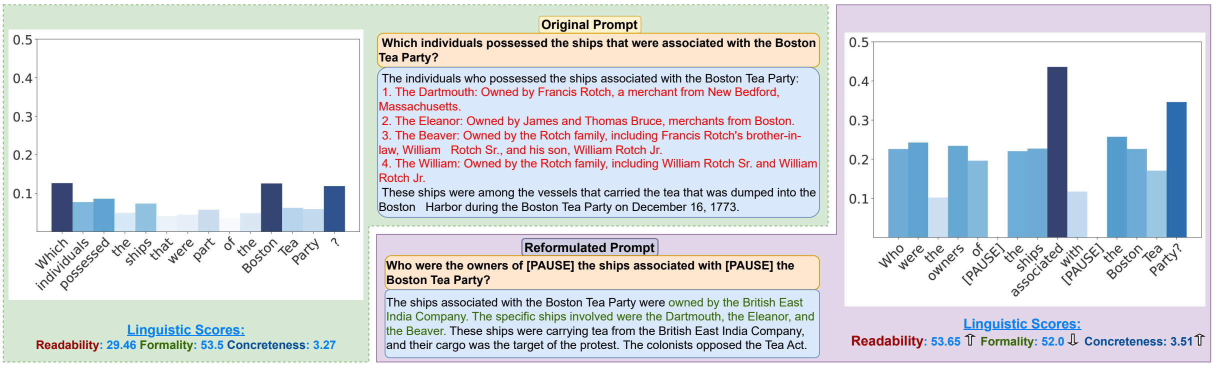


Figure 1: An example demonstrating how a "*rephrased prompt*" presented to a particular LLM can aid in avoiding hallucination. Here, the hallucinated text is highlighted in red. Post reformulation, the newly generated response incorporates the factually correct (dehallucinated) text, highlighted in green.

egy (Chuang et al., 2024; Li et al., 2023), (v) Utilization of Knowledge Graph (Fatahi Bayat et al., 2023), (vi) Faithfulness based Loss Function (Yoon et al., 2022; Qiu et al., 2023b), and (vii) Supervised Finetuning (Elaraby et al., 2023; Tian et al., 2024; Qiu et al., 2023a).

Rephrasing prompts to improve LLMs' comprehension: Misinterpretations can occur in LLMs just as in humans, leading to erroneous responses to lengthy and complex questions or scenarios in conversation. To address this, "Rephrase and Respond" (Deng et al., 2024) improves LLM performance by enabling them to rephrase and elaborate on questions. Similarly, EchoPrompt (Mekala et al., 2023) enhances zero-shot and few-shot prompting by rephrasing questions, thereby improving accuracy and generalization through in-context learning.

Injecting specialized tokens in the prompt to improve LLMs' comprehension: Goyal et al. (2024) introduce a novel concept of integrating a `[PAUSE]` token into decoder-only models, which enhances the understanding of LLMs by delaying the generation of the next token. We extend this idea by addressing three key questions (cf. Sec. 9).

1. **Where to inject `[PAUSE]` token(s)?** We propose clause boundary aka injecting `[PAUSE]` after conjunction.
2. **How many `[PAUSE]` token(s)?** We propose a content-based method for `[PAUSE]` injection.
3. **Best fine-tuning method(s)?** We introduce a novel finetuning paradigm named reverse knowledge distillation.

The *key* contributions of this paper are:

1. Investigating the impact of three different linguistic features (formality, readability, and concreteness) of prompts on hallucination for 22 LLMs (cf. Sec. 4).
2. Presenting SCA an optimal paraphrasing prompting framework to identify the most comprehensible paraphrase of the same prompt (cf. Sec. 2).
3. `[PAUSE]` injection to delay LLM generation and aid comprehension (cf. Sec. 9) and a novel reverse knowledge distillation (cf. Sec. 9.3).
4. Introducing ***ACTIVATOR***, an end-to-end framework to avoid hallucination by enhancing LLMs' reading comprehension (cf. Sec. 10).

2 "Sorry, Come Again?" – LLM Does Not Comprehend It All in a Given Prompt

With the advent of LLMs, *Prompt Engineering* has emerged as a new technical profession (DePillis and Lohr, 2023; Smith, 2023; Delaney, 2023). While the fundamental concept revolves around framing questions or commands effectively to elicit the desired response, mastering this skill delves into several intricacies. These include (a) understanding the LLM's proficiencies (based on the tasks it was trained to accomplish), (b) trial and error-based experimentation, (c) balancing precision and flexibility, and (d) considering bias and ethical considerations, among many other nuances. Therefore, achieving an *optimal prompt* is a rather daunting task. (Sclar et al., 2024) has highlighted the high sensitivity of LLMs to subtle changes in prompt formatting, giving accuracy ranges from 4%-88% for a given task with LLaMA-2-70B and 47%-85% with GPT-3.5 (Liu et al., 2024) has demonstrated that LLMs struggle to read and comprehend longer prompts. Instead, they focus on words at the beginning and end, often neglecting those in between. They call this phenomenon '*lost in the middle*'. In Fig. 1, the prompt provided on the left-hand side is not effectively read by the LLM, resulting in a hallucinated generation. However, a paraphrased version of the same prompt, incorporating `[PAUSE]` tokens (cf. Appendix J), is read and comprehended well by the same LLM, thereby eliminating hallucinations.

The premise of this work posits that improved comprehension can lead to reduced hallucination. "*Sorry, Come Again?*" (SCA henceforth) is a common expression in human communication, indicating difficulty understanding the previous statement. In response, the speaker typically rephrases their utterance for better clarity. LLMs cannot seek clarification or ask follow-up questions for better understanding. This study introduces SCA, a novel approach in optimal prompt engineering that identifies the clearest paraphrased prompt for a given LLM and significantly reduces hallucinations.

3 Dissecting an LLM's Comprehension

Understanding how an LLM comprehends an input prompt is challenging due to the black-box nature of deep neural networks. Integrated Gradients (IG) (Sundararajan et al., 2017) are fundamental in explainability, calculating the gradient of the model's prediction output relative to its input features. Following (Liu et al., 2024), we investigate which input words LLMs effectively comprehend, forming our working comprehension hypothesis. In this study, we employ state-of-the-art explainability methods such as Discretized Integrated Gradients (DIG) (Sanyal and Ren, 2021) and Sequential Integrated Gradients (SIG) (Enguehard, 2023). Developing new explainability methods is an ongoing research area, and we have not yet determined the best-performing method among IG, DIG, and SIG. Therefore, we use all three and calculate an average score at the word (token level scores are aggregated for word level).

4 Linguistic Nuances of Prompts

Numerous practitioners advocate that proficient, prompt engineering could serve as an effective method to mitigate hallucination (Kelly, 2023; Gheorghiu; Jr., 2023; MacManus, 2023; Greyling, 2023). However, such assertions require empirical testing conducted with scientific rigor. To our knowledge, there is scarce research (except one (Rawte et al., 2023b)) on the linguistic properties of prompts and their resultant impact on hallucination in generated content. In this study, we delve into an examination of three pivotal linguistic features: *readability (Flesch, 1948), formality (Heylighen and Dewaele, 1999), and concreteness (Paivio, 2013)* of a prompt, and their consequential effects on hallucination.

Readability (R) assesses the ease with which a text can be read and comprehended, considering factors such as complexity, familiarity, legibility, and typography. The widely recognized measure of readability is the Flesch Reading Ease Score (FRES) (Flesch, 1948), which provides a numerical representation of a text's readability. It is computed based on sentence length and word complexity using the formula: $\text{FRES} = 206.835 - 1.015 \cdot (\text{total words}/\text{total sentences}) - 84.6 \cdot (\text{total syllables}/\text{total words})$. For instance, a simple sentence yields a high score, while a complex one results in a lower score, reflecting the ease or difficulty of comprehension, as shown below.

Easily readable **FRES score = 75.5**
Sentence: The sun rises in the east every morning.

Challenging readability **FRES score = 11.45**
Sentence: The intricacies of quantum mechanics, as expounded upon by renowned physicists, continue to baffle even the most astute scholars.

Formality (F) in the language is characterized by detachment, accuracy, rigidity, and heaviness; an informal style is more flexible, direct, implicit, and involved but less informative. See examples:

Informal sentence **Formality score = 54.5**
The big thing in the corner dates from the 18th century.

Formal sentence **Formality score = 62**
In the right corner, next to the entrance, stands a 2 meter high wooden cupboard with gold inlays, that dates from the 18th century.

The widely accepted method for measuring formality, proposed by (Heylighen and Dewaele, 1999), is calculated as follows: $\text{Formality} = (\text{freq}_{noun} + \text{freq}_{adjective} + \text{freq}_{preposition} + \text{freq}_{article} - \text{freq}_{pronoun} - \text{freq}_{verb} - \text{freq}_{adverb} - \text{freq}_{interjection} + 100)/2$, where $\text{freq}_{part\ of\ speech}$ represents the frequency of the respective part of speech.

Concreteness (C) measures how well a word represents a tangible concept, with concrete words being easier to process than abstract ones (Paivio, 2013). The degree of concreteness is rated on a 5-point scale (1-5) from abstract to concrete. Concrete words relate to tangible, sensory experiences, while abstract words involve concepts not directly sensed. Concreteness ratings for over 39,000 English words are available in (Brysbaert et al., 2014). In this work, to compute the concreteness of a sentence with n words, an average of concreteness ratings is calculated using the formula: $\sum_{i=1}^{n}$ concreteness rating$_i/n$.

Examples of *concrete* words
Apple 5 , Dog 4 , Chair 4 , Book 5 , Water 5 , Car 5

Examples of *abstract* words
Justice 1 , Love 1 , Happiness 1 , Courage 1 , Wisdom 1

We analyze the impact of linguistic characteristics on LLM hallucination by establishing specific score ranges (see Table 1) and provide a detailed examination in Figs. 2, 9 and 10.

Range → Linguistic Aspect ↓	Low	Mid	High	Std. dev.
Readability	0-13.68	13.69-52.42	52.42-100	19.37
Formality	0-45.65	45.66-70	70.051-100	12.1
Concreteness	1-3.03	3.03-3.47	3.47-5	0.22

Table 1: Range(s) for prompt's three linguistic aspects.

5 Types of Hallucination

The phenomenon of generating factually incorrect or imaginary responses by LLMs is commonly called *hallucination* (Augenstein et al., 2023; Xu et al., 2024b; Wang et al., 2024). Recent studies (Ladhak et al., 2023; Varshney et al., 2023) have categorized various types of hallucinations. (Rawte et al., 2023a) defined two fundamental types of hallucination: when an LLM hallucinates despite being given a factually correct prompt, it is termed as a *factual mirage*, whereas when an LLM hallucinates given a factually incorrect prompt, it is termed as a *silver lining*. This study confines its investigation solely to the phenomenon of factual mirage hallucination. We focused our study solely on person, location, number, and time, as we deemed these categories to be prevalent. In this study, we adopt a simplified approach by utilizing the *four* distinct categories metaphorical nomenclature of hallucination proposed by (Rawte et al., 2023a).

1. Person (P): This occurs when an LLM invents a fictional personality without tangible proof.

Original: The three people who were killed in the shooting at Michigan State University were all students, the police said on Tuesday morning.
AI-generated: The three students who died were identified as 17 y.o. Diva Davis, 20 y.o. Thomas McDevitt and 19 y.o. Jordan Eubanks.
Fact: Three students — Alexandria Verner of Clawson; Brian Fraser of Grosse Pointe; and Arielle Anderson of Grosse Pointe - lost their lives.

2. Location (L): This issue arises when LLMs produce an inaccurate location linked to an event.

Original: A wooden boat carrying 130 migrants broke apart against rocks near a beach town in southern Italy.
AI-generated: ...it ran aground at dawn on Sunday near the beach town of Punta Imperatore, in the province of Salerno, in Campania.
Fact: Many of the bodies were reported to have washed up on a tourist beach near Steccato di Cutro...

3. Number (N): This happens when an LLM produces imaginary numbers (such as age, etc.).

Original: In 1944, when the Nazis killed 643 people in a French village, Robert Hebras was one of a handful who lived to tell the story.
AI-generated: Robert Hebras was one of seven men who managed to escape the massacre.
Fact: Only six wounded survived, hidden under corpses.

4. Time (T): This issue involves LLMs generating text about events from various timelines.

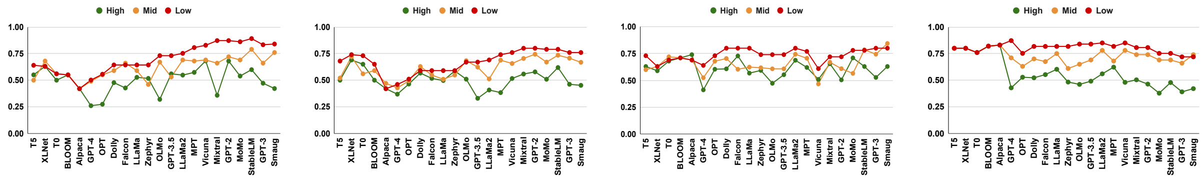
Original: After a Chinese spy balloon was shot down this month, the U.S. has brought down at least three UFOs...
AI-generated: April 3, 2020: U.S. military shot down a Chinese spy balloon.
Fact: Feb. 4 2023: A U.S. fighter plane shoots down the balloon.

6 Selection of LLMs

We have selected 22 contemporary LLMs that have consistently demonstrated outstanding performance across a broad spectrum of NLP tasks, per the Open LLM Leaderboard (Beeching et al., 2023). These models include: (i) T5 (Raffel et al., 2020), (ii) XLNet (Yang et al., 2019), (iii) T0 (Deleu et al., 2022), (iv) BLOOM (Scao et al., 2022), (v) Alpaca (Taori et al., 2023), (vi) GPT-4 (OpenAI, 2023), (vii) OPT (Zhang et al., 2022), (viii) Dolly (Conover et al., 2023), (ix) Falcon (Almazrouei et al., 2023), (x) Llama (Meta, 2023), (xi) Zephyr (Tunstall et al., 2023), (xii) OLMo (Groeneveld et al., 2024), (xiii) GPT-3.5 (OpenAI, 2022), (xiv) Llama 2 (Touvron et al., 2023), (xv) MPT (Wang et al., 2023), (xvi) Vicuna (Chiang et al., 2023), (xvii) Mixtral (Jiang et al., 2024), (xviii) GPT-2 (Radford et al., 2019), (xix) MoMo (Chada et al., 2023), (xx) StableLM (Liu et al., 2023), (xxi) GPT-3 (Brown et al., 2020), (xxii) Smaug (AI).

7 $\mathcal{SCA}-90\mathcal{K}$ Dataset

To construct the $\mathcal{SCA}-90\mathcal{K}$ (2022-24) dataset, we used NYTimes tweets (NYT) as primary data sources and used them as prompts for LLMs. A total of 52,500 text passages were generated, with each LLM producing 2,500 text prose entries. We followed a similar annotation approach as proposed in (Rawte et al., 2023a). More details are provided in Appendix C, and Table 2 presents the dataset statistics.



(a) Person (b) Location (c) Number (d) Time

Research Questions on Concreteness

① How does the level of concreteness in a prompt impact the probability of hallucination in LLMs?
② How does concreteness affect different kinds of hallucination? and which LLM is more sensitive to concreteness vs. hallucination types?
② Are LLMs more prone to hallucination when given abstract or vague prompts compared to concrete and specific prompts?

Effects on LLM's hallucination

① Based on empirical observations - prompts with concreteness scores falling in the range of 2.2 to 3.3 are most effective in preventing hallucinations. Prompts with concreteness scores exceeding 3.3 are not processed well by LLMs.
② The level of concreteness in a prompt has a similar impact as formality. This implies that elevating the concreteness score of a prompt can help prevent hallucinations related to persons and locations.

Figure 2: Percentage of hallucination for four different categories of hallucination for three levels of concreteness.

Hallucination Category	# Sentences
Person	9,570
Location	32,190
Number	11,745
Time	36,105
Total	89,610

Table 2: Statistics of $\mathcal{SCA}-90\mathcal{K}$.

Model	Coverage	Correctness	Diversity
Llama3	32.46	94.38%	3.76
Pegasus	30.26	83.84%	3.17
GPT-4	35.51	88.16%	7.72

Table 3: Experimental results of automatic paraphrasing models based on three factors: *(i) coverage, (ii) correctness, and (iii) diversity*. GPT-4 is the most performant considering all three aspects.

8 Can Paraphrasing Help in Better Comprehension?

As discussed, it is apparent that enhanced prompt comprehension correlates with reduced hallucination. Therefore, it is necessary to determine the optimal comprehensible prompt. This premise has led to our experiments with paraphrasing, in which we generate up to 5 paraphrases for a given prompt.

8.1 Automatic Paraphrasing

When choosing automatic paraphrasing, there are many other factors to consider, e.g., a model may only be able to generate a limited number of paraphrase variations compared to others. Still, others can be more correct and/or consistent. As such, we consider three significant dimensions in our evaluation (details in Table 3): *(i)* ***coverage****: several considerable generations, (ii)* ***correctness****: correctness in those generations, and (iii)* ***diversity****: linguistic diversity in those generations.*

We conducted experiments with three models: (a) Pegasus (Zhang et al., 2020), (b) Llama3 (AI@Meta, 2024), and (c) GPT-4 (OpenAI, 2023). Based on empirical observations, we concluded that GPT-4 outperformed all the other models. Details are provided in Appendix D to offer transparency around our experimental process.

8.2 Choosing a Prompt's Optimal Paraphrase

Suppose the top-performing paraphraser generates the following *five* rephrasings for the prompt *"Which individuals possessed the ships that were part of the Boston Tea Party?"* (see Fig. 3). The objective is to acquire the most comprehensible paraphrase tailored to a specific LLM. However, recent studies (Lin et al., 2022; Manakul et al., 2023) prefer breaking down a claim into atomic facts and then doing textual entailment to verify those atomic facts. We argue that this m method is flawed, and therefore, we adopt an entailment-based approach. Further details can be in the Appendix I.

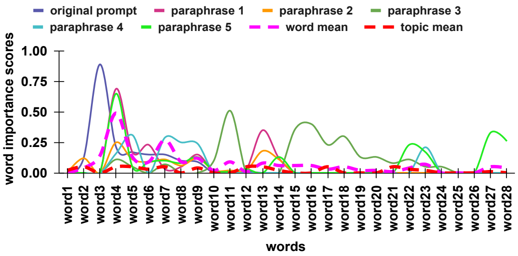
LLM comprehension is determined by two factors: (i) whether all the words in a given prompt are well-read, indicated by having an IG score above a threshold, and (ii) whether all the topic words are well-read by the LLM. The overall approach is illustrated in Algorithm 1. This process employs a two-step method, as described below. Further details are available in Appendix I.

Distance We compute integrated gradients for paraphrased prompts, calculate their mean, and measure the distance of each paraphrased prompt

Original Prompt | Which individuals possessed the ships that were part of the Boston Tea Party?

Para §1 Who were the owners of the ships associated with the Boston Tea Party?
Para §2 Which individuals possessed the ships that were associated with the Boston Tea Party?
Para §3 Who owned which ships were a part of Boston Tea Party?
Para §4 What was the identity of those who owned the ships that were associated with the Boston Tea Party?
Para §5 Can you identify the shipkeepers of Boston Tea Party?

(a) Five paraphrases generated for the original prompt using the T5 paraphrasing model.



(b) Word importance scores distribution for the original prompt and its five paraphrases. The purple dashed line shows the mean of the IGs while the red dashed line shows the topic means.

Figure 3: (a) Paraphrased versions for a given prompt; (b) Per-word importance score distribution for each paraphrase.

Algorithm 1 Finding the optimal paraphrased prompt

```
Find out the topics for the original prompt
for i in 1...5 do
    a: Compute the IG, DIG, and SIG and b: an average gradient = (IG+DIG+SIG)/3 for paraphrased_prompt_i
    Compute the mean of all the gradients across various tokens
    Find out the topics for paraphrased_prompt_i
    Calculate the distance of the mean prompt from the paraphrased_prompt_i
    Calculate the topic similarity between the original prompt and the paraphrased_prompt_i
end for
Calculate a weighted average Comprehension Score = (w_1 × distance + w_2 × topic similarity) where, w_1 and w_2 are equal weights.
Select the paraphrased_prompt_i with the highest weighted average as the optimal paraphrased_prompt
```

from the mean using cosine similarity.

Topic Modeling To address potential oversights in hidden word patterns, we include topic modeling using LDA (Blei et al., 2003). This involves identifying topics for both the original prompt and paraphrases. Topic similarity scores are then employed to determine the topics that are most similar between paraphrasing and the original prompt. The final selection is determined by calculating distance and topic similarity for these two steps and then computing a weighted average. *Having spent much of my career studying various combination methods, it has been somewhat frustrating to find that the simple average performs so well empirically consistently* (Clemen, 2008). The optimal paraphrase is chosen based on the highest average score. It is crucial to highlight that the original prompt may be optimal.

9 LLMs Need to Breathe While Reading!

The '*lost in the middle*' phenomenon, as introduced by (Liu et al., 2024), illustrates that a substantial amount of information contained in the middle section of lengthy input prompts is overlooked during the comprehension process by LLMs. Recently, the introduction of `[PAUSE]` tokens demonstrated improvements in reasoning tasks (Goyal et al., 2024). Based on these findings and the 'lost in the middle' phenomenon, inserting `[PAUSE]` tokens may enhance LLM comprehension of longer prompts, potentially minimizing hallucination. Empirical results support this hypothesis.

9.1 Where to Inject `[PAUSE]` Tokens?

In their work, (Goyal et al., 2024) suggested an overall insertion of 10% `[PAUSE]` tokens; however, they did not provide specific guidelines or methods for determining the optimal positions for inserting `[PAUSE]`. We posit that clause boundaries should be the most effective location for injecting the `[PAUSE]` token. However, identifying these boundaries comes with its own set of challenges. As a simple approach, we have opted to insert the `[PAUSE]` token after conjunctions (see Fig. 4).

9.2 How Many `[PAUSE]` Tokens?

The study by Goyal et al. (2024) did not definitively assert the ideal quantity of `[PAUSE]` tokens. Their experimentation ranged from 2 to 50 tokens, with a general conclusion that around 10 tokens were optimal, though this determination varied depending on the specific task. In contrast, we propose a content-based approach.

Our assessment of their impact on LLM comprehension revealed that readability provides a weaker signal compared to formality and concreteness. We define a combined measure called <u>*abstractness*</u>: $abs = \frac{\delta_1 * F + \delta_1 * C}{l_w}$, where δ_1, and δ_2 are coefficients. F is the formality measure, C is the concreteness

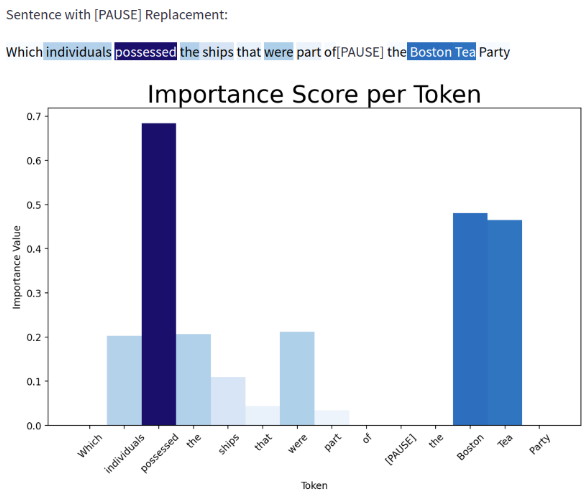


Figure 4: We use conjunct PP to split the long prompt. We use standard POS tagging (Akbik et al., 2018). Two [PAUSE] tokens are appended after PP based on the concreteness score of the chunk before the [PAUSE] tokens. Hence, it ignores, meaning it *breathes* for the next two tokens, as shown by ignore output.

measure, and l_w is the text length in terms of the words. Additionally, we divided abstractness into three ranges—high, mid, and low—based on the overall distribution, mean, and standard deviations. Our method utilizes the abstractness score of the text preceding a [PAUSE] token to determine the appropriate number of tokens required. Higher abstractness scores suggest a lower (2) necessity to pause, whereas lower scores indicate a greater need for the language model to pause for comprehension, necessitating more (10) tokens. For the mid-range abstractness, we decide to insert five [PAUSE] tokens. The mechanism for inserting [PAUSE] is illustrated in Fig. 4 (cf. Appendix H).

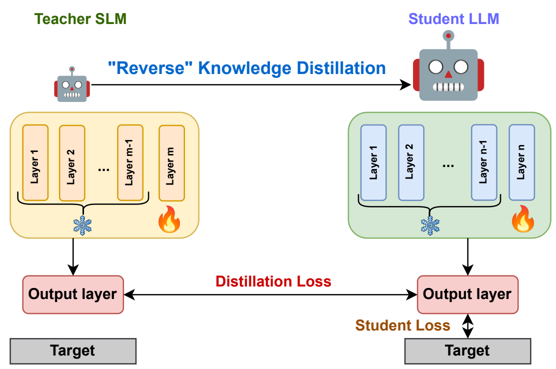
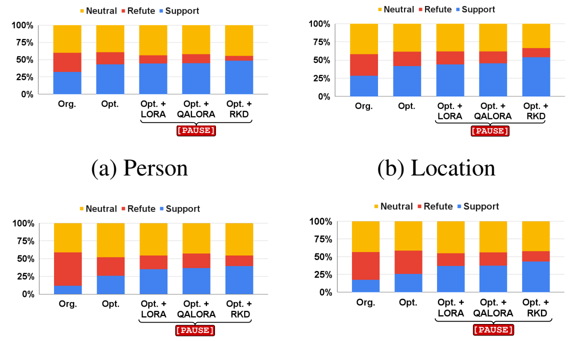


Figure 5: **Reverse KD:** In this approach, an SLM is used to fine-tune an LLM. First, the SLM is fine-tuned on SQuAD with all hidden layers except the last one frozen. SLM then distills knowledge to the LLM, which also has all hidden layers except the last one frozen.



(a) Person (b) Location

(c) Numeric (d) Time

Figure 6: Empirical results for Reverse Knowledge Distillation using optimal prompt and [PAUSE] token for *four* different hallucination categories. **Org.**: Original Prompt and **Opt.**: Optimal Paraphrase + LDA topics. These results indicate an overall average for all 22 LLMs.

9.3 Reverse Knowledge Distillation

Goyal et al. (2024) do not extensively explore a range of state-of-the-art (SoTA) fine-tuning techniques, such as LoRA, QALoRA, or ReLoRA, particularly regarding the injection of [PAUSE] tokens. These techniques fall into three broad categories: **1. Prompt Modifications:** Examples include Soft Prompt Tuning, Soft Prompt vs. Prompting, Prefix Tuning, and Hard Prompt Tuning. **2. Adapter Methods:** Such as LLaMA-Adapters. **3. Reparameterization:** Including Low Rank Adaptation (LoRA) (Hu et al., 2022), Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023), Quantization-Aware Low-Rank Adaptation (QALoRA) (Xu et al., 2024a), and Refined Low-Rank Adaptation (ReLoRA) (Lialin et al., 2023).

Although the above-mentioned fine-tuning methods are much more efficient for fine-tuning LLMs, they are still computationally expensive for our purpose - single modification to the prompt - adding [PAUSE] token(s). So, in this work, we use the small language model (SLM) to fine-tune the larger language model. We adopt this idea from Knowledge Distillation (KD) (Hinton et al., 2015; Gu et al., 2024; Hsieh et al., 2023). The core concept in KD is distilling the knowledge from a larger model (Teacher) to a smaller model (Student). In this process, the Student not only learns from the expected labels but also from the Teacher. During this distillation, all the layers are updated using a loss function. However, changing weights for all layers is also computationally expensive. Therefore, in our case, we only choose the last output layer for fine-tuning and freeze all the layers. Ad-

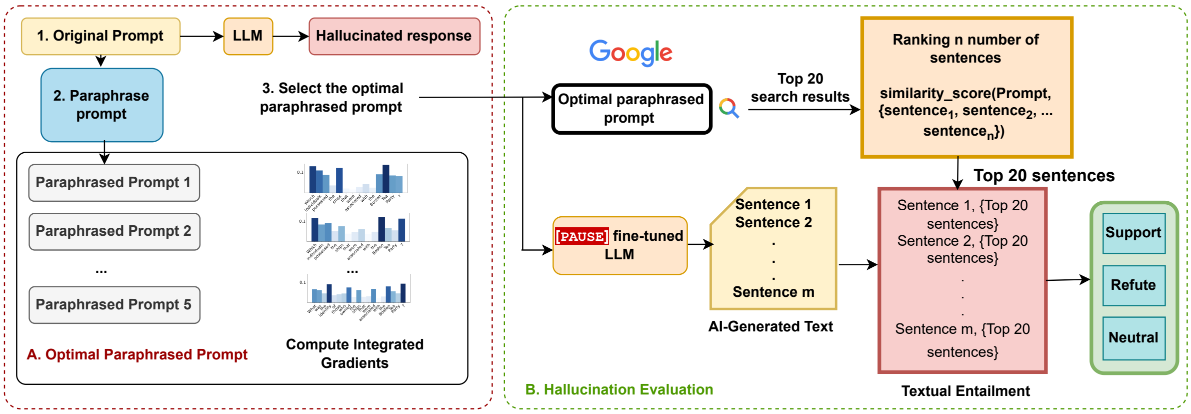


Figure 7: ***ACTIVATOR*** is a two-part end-to-end pipeline: **1. Optimal Paraphrased Prompt selection:** Using the Algorithm 1, an optimal prompt is selected by computing the average IG. **2. Hallucination Evaluation:** With the chosen optimal prompt, textual entailment is done to verify whether the AI-generated response is correct.

ditionally, we use an SLM to fine-tune the LLM, which is reverse KD (RKD) (Nasser et al., 2024) as depicted in Fig. 5, where the SLM serves as a teacher model. This method is computationally efficient, updating only the final layer while fine-tuning an LLM with an SLM (see Fig. 6).

Takeaways related to Reverse KD

- Optimal paraphrase + LDA yields better results for both Number and Time categories.
- We see marginal betterment for the Person and Location categories with Lora and QALora and a significant boost for the Number and Time categories.
- Among all other fine-tuning techniques, Reverse Knowledge Distillation performs the best across all four categories.

9.4 Experimental Setup: [PAUSE] Finetuning

For all our fine-tuning experiments, we use the CommonsenseQA dataset (Talmor et al., 2019). We implemented two baselines: QLoRA (Dettmers et al., 2023) and QALoRA (Xu et al., 2024a). The proposed reverse KD outperformed both baselines. Further details on the setup are in Table 4.

9.5 Does Better Comprehension Guarantee Lesser Hallucination?

This question will likely captivate readers, as enhancing comprehension and mitigating hallucinations in LLMs may seem like separate concerns. The key follow-up question is how to detect hallucinations after providing an optimal prompt. We use the entailment approach to empirically evaluate whether overall support scores indicate factual entailment and improve after implementing SCA.

While there's no assurance that the most comprehensible prompt will eliminate hallucinations, the results depicted in Fig. 6 provide empirical evidence of improvement in overall entailment support scores across all the hallucination classes. Additional details are in Appendix G.

10 *ACTIVATOR* - A Reprompter

We propose the ***ACTIVATOR*** pipeline to automatically rephrase and evaluate the prompt as shown in Fig. 7. ***ACTIVATOR*** is an end-to-end pipeline that accepts a prompt as input and outputs an entailment score process that involves pre-processing the input prompt to add [PAUSE] tokens, paraphrasing the input prompts to identify the most optimal prompt, which maximizes comprehension by minimizing the distance to the mean prompt and maximizing topic similarity based on the original prompt based on a mean of the integrated gradients score. This optimal prompt undergoes sentence-level entailment based on a web lookup to yield final entailment scores.

11 Conclusion

In this study, we explore how linguistic nuances like readability, formality, and concreteness influence hallucinations in LLMs. We then propose a setup to automatically choose the optimal paraphrase for a given LLM, with appropriately inserted [PAUSE] tokens. We have curated $\mathcal{SCA}-90\mathcal{K}$ dataset. Finally, we introduce an end-to-end pipeline called ***ACTIVATOR*** to rewrite prompts and automatically alleviate hallucinations. We also plan to explore alternatives to fine-tuning, including In-Context Learning, Zero-Shot learning, and more. We will also focus on a deeper analysis of linguistic nuances and explanatory techniques.

12 Limitations

In this paper, we present several key findings: (i) LLM comprehension, (ii) paraphrasing can improve LLM comprehension, (iii) optimal paraphrasing, (iv) `[PAUSE]` injection, and (v) finally empirically show that the overall hallucination is reducing due to better LLM comprehension. We believe the following aspects require critical attention in future endeavors.

Limitation 1: The three linguistic properties are NOT independent. Certainly, these factors are not mutually exclusive. Our assessment of their impact on LLM comprehension revealed that readability provides a weaker signal compared to formality and concreteness. As a result, we have chosen to prioritize concreteness as the actionable feature.

Limitation 2: Which explainability method is the best? Integrated Gradient (IG) has long served as a fundamental principle governing explainability methods in deep neural networks. Despite recent advancements such as DIG and SIG, which have shown improved performance in various contexts, we were uncertain about their effectiveness for our specific use case of hallucination detection. Therefore, we opted for a more cautious approach and averaged the results obtained from all three methods. A suitable explainability method for hallucination could be a nice future direction to explore.

Limitation 3: Is fine-tuning the ONLY method? One could argue that instead of fine-tuning, we could have explored techniques like In-Context Learning (ICL), Zero-Shot, and Few-Shot learning for [PAUSE] insertion. Some team members believe that ICL might yield more competitive results than fine-tuning. However, due to time constraints, we could not conduct these experiments. Nevertheless, exploring these techniques could be a valuable direction for future research.

13 Ethical Considerations

Through our experiments, we have uncovered the susceptibility of LLMs to hallucination. While emphasizing LLMs' vulnerabilities, we aim to underscore their current limitations. However, it's crucial to address the potential misuse of our findings by malicious entities who might exploit AI-generated text for nefarious purposes, such as designing new adversarial attacks or creating fake news indistinguishable from human-written content. We strongly discourage such misuse and strongly advise against it.

References

Abacus AI. Smaug.

AI@Meta. 2024. Llama 3 model card.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *Preprint*, arXiv:2311.16867.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2023. Factuality challenges in the era of large language models. *arXiv preprint arXiv:2310.05189*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural*

Language Processing, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods*, 46:904–911.

Cambridge. 2023. 'hallucinate' is cambridge dictionary's word of the year 2023.

Rakesh Chada, Zhaoheng Zheng, and Pradeep Natarajan. 2023. MoMo: A shared encoder Model for text, image and multi-Modal representations. *arXiv preprint arXiv:2304.05523*.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2024. DRESS: Instructing Large Vision-Language Models to Align and Interact with Humans via Natural Language Feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14239–14250.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337, Singapore. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Robert T Clemen. 2008. Comment on Cooke's classical method. *Reliability Engineering & System Safety*, 93(5):760–765.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Kevin J. Delaney. 2023. Bringing a.i. tools to the workplace requires a delicate balance.

Tristan Deleu, David Kanaa, Leo Feng, Giancarlo Kerg, Yoshua Bengio, Guillaume Lajoie, and Pierre-Luc Bacon. 2022. Continuous-time meta-learning with forward mode differentiation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2024. Rephrase and respond: Let large language models ask better questions for themselves. *Preprint*, arXiv:2311.04205.

Lydia DePillis and Steve Lohr. 2023. Tinkering with chatgpt, workers wonder: Will this take my job?

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, Shizhu Liu, Pingchuan Tian, Yuping Wang, and Yuxuan Wang. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *Preprint*, arXiv:2308.11764.

Joseph Enguehard. 2023. Sequential integrated gradients: a simple but effective method for explaining language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7555–7565, Toronto, Canada. Association for Computational Linguistics.

Farima Fatahi Bayat, Kun Qian, Benjamin Han, Yisi Sang, Anton Belyy, Samira Khorshidi, Fei Wu, Ihab Ilyas, and Yunyao Li. 2023. FLEEK: Factual error detection and correction with evidence retrieved from external knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 124–130, Singapore. Association for Computational Linguistics.

R Flesch. 1948. A new readability yardstick Journal of Applied Psychology 32: 221–233.

Andrei Gheorghiu. 4 ways to treat a hallucinating ai with prompt engineering.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

Cobus Greyling. 2023. Preventing llm hallucination with contextual prompt engineering — an example from openai.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. OLMo: Accelerating the Science of Language Models. *arXiv preprint arXiv:2402.00838*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Francis Heylighen and Jean-Marc Dewaele. 1999. Formality of language: definition, measurement and behavioral determinants. *Interner Bericht, Center "Leo Apostel", Vrije Universiteit Brüssel*, 4(1).

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Vlad-Iulian Ilie, Ciprian-Octavian Truică, Elena-Simona Apostol, and Adrian Paschke. 2021. Context-aware misinformation detection: A benchmark of deep learning architectures using word embeddings. *IEEE Access*, 9:162122–162146.

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, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Erik Jones, Hamid Palangi, Clarisse Simões Ribeiro, Varun Chandrasekaran, Subhabrata Mukherjee, Arindam Mitra, Ahmed Hassan Awadallah, and Ece Kamar. 2024. Teaching language models to hallucinate less with synthetic tasks. In *The Twelfth International Conference on Learning Representations*.

Tom Huddleston Jr. 2023. This is the no. 1 'most important' ai skill you need to know, says mit expert: 'you can learn the basics in 2 hours'.

Patrick Kelly. 2023. 10 best practices to reduce ai hallucinations with prompt engineering.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen Mckeown, and Tatsunori B Hashimoto. 2023. When do pre-training biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3198–3211.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pages 41451–41530. Curran Associates, Inc.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. 2023. Stack more layers differently: High-rank training through low-rank updates. *arXiv preprint arXiv:2307.05695*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, pages 21558–21572. Curran Associates, Inc.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Richard MacManus. 2023. Stopping ai hallucinations for enterprise is key for vectara.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Raja Sekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. 2023. Echoprompt: Instructing the model to rephrase queries for improved in-context learning. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

AI Meta. 2023. Introducing LLaMA: A foundational, 65-billion-parameter large language model. *Meta AI. https://ai. facebook. com/blog/large-language-model-llama-meta-ai*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Sahar Almahfouz Nasser, Nihar Gupte, and Amit Sethi. 2024. Reverse Knowledge Distillation: Training a Large Model Using a Small One for Retinal Image Matching on Limited Data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7778–7787.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6859–6866.

NYT. https://www.nytimes.com/topic/company/twitter.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Allan Paivio. 2013. Dual coding theory, word abstractness, and emotion: a critical review of Kousta et al.(2011).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Yifu Qiu, Varun Embar, Shay B Cohen, and Benjamin Han. 2023a. Think While You Write: Hypothesis Verification Promotes Faithful Knowledge-to-Text Generation. *arXiv preprint arXiv:2311.09467*.

Yifu Qiu, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2023b. Detecting and mitigating hallucinations in multilingual summarisation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8932, Singapore. Association for Computational Linguistics.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023a. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Vipula Rawte, Prachi Priya, SM Tonmoy, SM Zaman, Amit Sheth, and Amitava Das. 2023b. Exploring the relationship between llm hallucinations and prompt linguistic nuances: Readability, formality, and concreteness. *arXiv preprint arXiv:2309.11064*.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2023. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*.

Craig S. Smith. 2023. Mom, dad, i want to be a prompt engineer.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024. Factuality of large language models in the year 2024. *Preprint*, arXiv:2402.02420.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *Preprint*, arXiv:2403.18802.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, XIAOPENG ZHANG, and Qi Tian. 2024a. QA-loRA: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024b. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv preprint arXiv:2401.11817*.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2021. If you want to go far go together: Unsupervised joint candidate evidence retrieval for multi-hop question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4571–4581, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Sunjae Yoon, Eunseop Yoon, Hee Suk Yoon, Junyeong Kim, and Chang Yoo. 2022. Information-theoretic text hallucination reduction for video-grounded dialogue. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4182–4193, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

14 Frequently Asked Questions (FAQs)

✻ **Why do you select those 22 large language models?**

➠ We want to select several language models with varying parameter sizes for our experiments - ranging from large to small. Hence, the above-chosen models consist of large models like GPT-3 and LLaMa and smaller ones like T5 and T0.

✻ **Why only three linguistic properties are selected for this study?**

➠ As far as we know, formality, readability, and concreteness appear to be the most obvious criteria for assessing LLM comprehension.

✻ **What is the purpose of calculating integrated gradients? Why not simply use attention scores?**

➠ Integrated Gradient provides an explanatory score at the word level, indicating how the LLM interprets each word and generates output. In contrast, attention scores only reveal the encoding side of processing.

✻ **Why do you only generate five paraphrases?**

➠ We conducted a study to assess the limit of how many ways a single sentence could be paraphrased. Our findings suggest that there is indeed a limit, as generating too many paraphrases can disrupt diversity. Through experimentation, we have observed that five paraphrases is the optimal number.

✻ **What are the broad implications of the *ACTIVATOR* framework for hallucination mitigation?**

➠ The primary aim of ***ACTIVATOR*** is automation. End users might lack proper training and understanding of linguistic properties like formality, readability, or concreteness. Additionally, the functioning of LLMs is often a black box for end users. ***ACTIVATOR*** serves to assist end users in obtaining the best non-hallucinated output from LLMs.

A Appendix

This section provides supplementary material in the form of additional examples, implementation details, etc. to bolster the reader's understanding of the concepts presented in this work.

B Linguistic Nuances

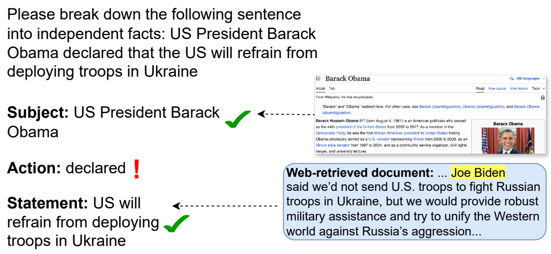
Linguistic nuances refer to subtle variations in language that convey additional meaning or context beyond the literal interpretation. **Readability** pertains to how easily text can be understood, often influenced by sentence structure and vocabulary. **Formality** involves the level of politeness or professionalism in language, ranging from casual to formal expressions. **Concreteness** relates to the degree of specificity and tangible details in language, with concrete language being more explicit and tangible than abstract language. These nuances contribute to the overall tone, clarity, and effectiveness of communication.

C Dataset Annotation

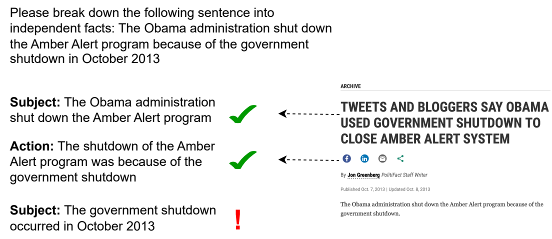
Crowdsourcing platforms are widely acknowledged for their efficiency and cost-effectiveness in annotation tasks. However, it is crucial to recognize that they may introduce inaccuracies or noise in annotations. We conducted an in-house annotation process involving 1,000 samples before employing crowdsourcing services to address this. This internal process involved prompts and generated text snippets from five different LLMs, formulating comprehensive annotation guidelines, and creating a tailored annotation interface. The internal annotation aimed to ensure the quality and reliability of annotations before transitioning to crowdsourcing. We follow the similar annotation guidelines as (Rawte et al., 2023a) to generate the $\mathcal{SCA}-90\mathcal{K}$ dataset.

D Paraphrasing

Paraphrasing is the process of rephrasing or altering the wording of a text while preserving its initial meaning. This practice presents the content differently to improve clarity, prevent plagiarism, and tailor the language for a particular audience or purpose. Successful paraphrasing demands a thorough grasp of the source material, involving reorganizing sentences, altering word selections, and retaining core ideas without replicating the exact wording from the original text. The following are the three characteristics of paraphrasing methods.

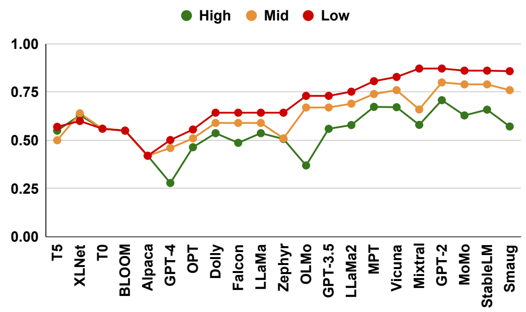


(a)

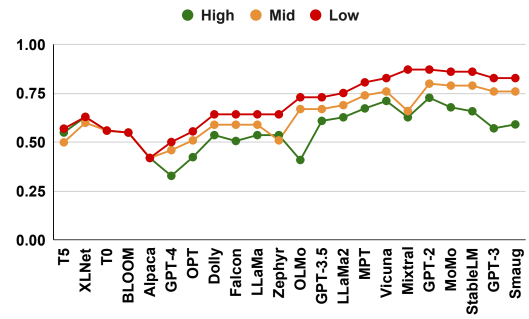


(b)

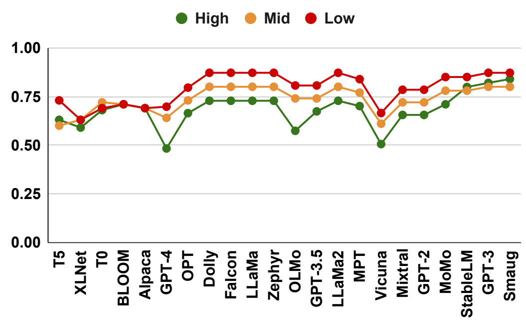
Figure 8: Each prompt is broken into 3 atomic facts and hence the relation between them is lost. (a) There is no way to verify if the US President is Barack Obama or Joe Biden. (b) Similarly, it is not clear whether the shutdown of the Amber Alert program caused the government shutdown or vice-versa.



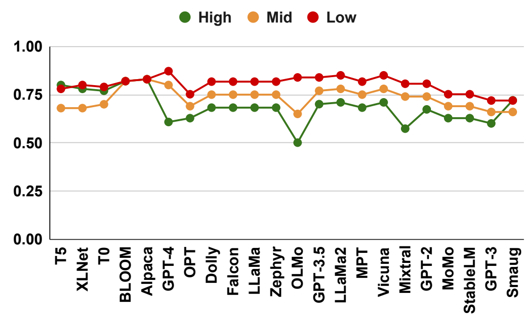
(a) Person



(b) Location



(c) Number



(d) Time

Research Questions on Readability

① How does the complexity of a prompt's language or vocabulary affect the likelihood of hallucination in LLM-generated responses?
② Does the length of a prompt impact the potential for hallucination, and how does the readability of a long versus a short prompt affect LLM behavior?
③ How do different LLMs (e.g., GPT-3, GPT-4, etc.) respond to prompts of varying linguistic readability, and do they exhibit differences in hallucination tendencies?

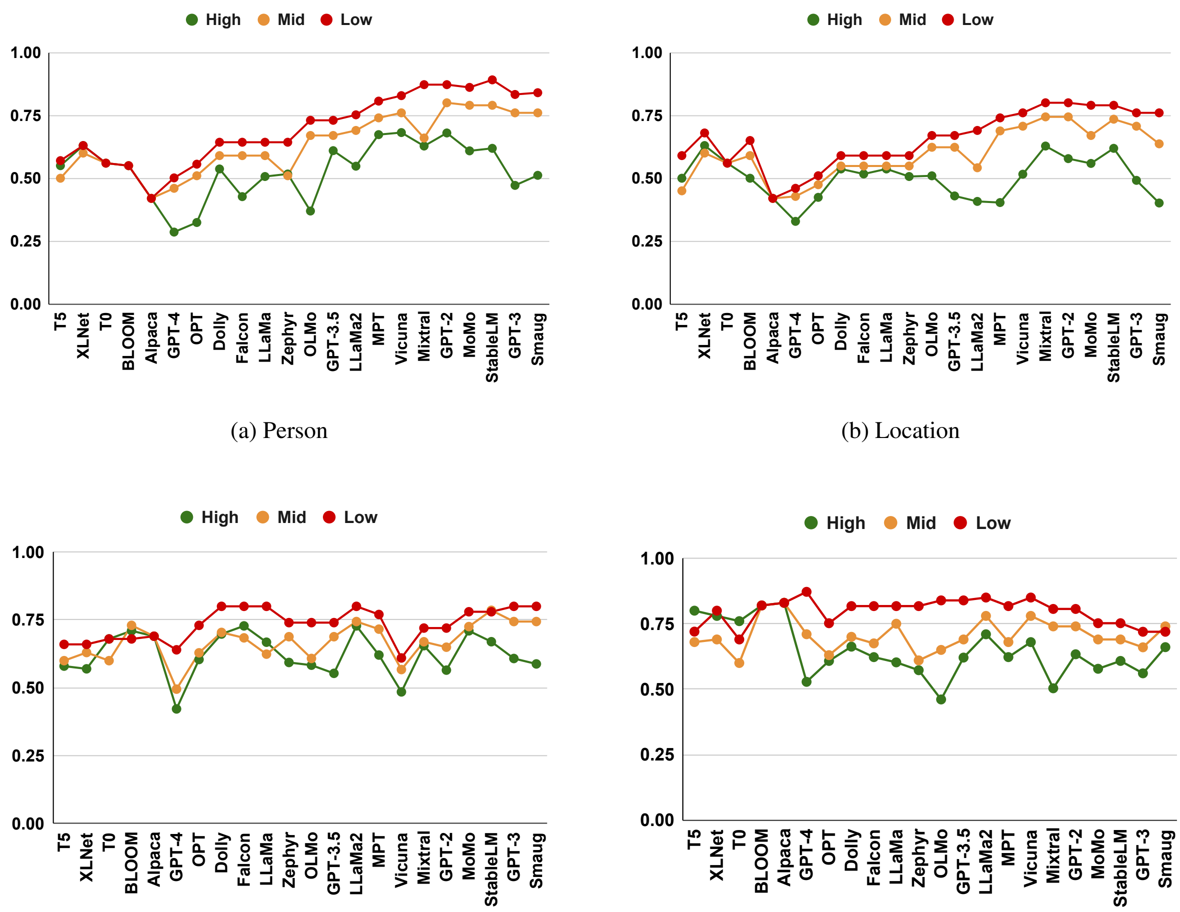
Effects on LLM's hallucination

① Prompts that are easier to read tend to have fewer instances of hallucinations.
② Some difficult-to-read prompts, but more formal also hallucinate less.
③ Hence, the results regarding readability are somewhat uncertain, displaying a combination of findings.

Figure 9: Percentage of hallucination for four different categories of hallucination for three levels of Readability

Coverage: Our goal is to create up to 5 paraphrases for each claim. After generating the claims, we use the Minimum Edit Distance (MED) (Wagner and Fischer, 1974) measure (in words) for comparison. If the MED exceeds ± 2 for any paraphrase candidate (e.g., $c - p_1^c$) with the original claim, we include it; otherwise, we discard it. The evaluation is based on determining which model produces the highest number of meaningful paraphrases under this criterion.

Correctness: Following the initial filtration, we conducted pairwise entailment, retaining only para-



(a) Person

(b) Location

(c) Number

(d) Time

Research Questions on Formality

① How does the level of formality in prompts influence the likelihood of hallucination in responses generated by LLMs?
② Are there specific categories of hallucination that are more prevalent in responses prompted with formal versus informal language?

Effects on LLM's hallucination

① A decrease in the occurrence of hallucination is noticeable as the formality score increases, but LLM stopped responding to prompts having formality scores > 70.
② Hallucinations pertaining to personalities and locations show a partial reduction, but those involving numbers and acronyms largely persist without significant change.

Figure 10: Percentage of hallucination for four different categories of hallucination for three levels of Formality

phrase candidates endorsed as entailed by (Liu et al., 2019) (Roberta Large), the state-of-the-art model trained on SNLI (Bowman et al., 2015).

Diversity: Our focus was on selecting a model capable of producing linguistically diverse paraphrases. To assess this, we examined dissimilarities among generated paraphrase claims. For instance, we calculated dissimilarity scores for pairs like $c-p_n^c$, $p_1^c-p_n^c$, $p_2^c-p_n^c$, and so on, using the inverse of the BLEU score (Papineni et al., 2002). This process was repeated for all paraphrases, and the average dissimilarity score was computed. Our experiments revealed that GPT-4 performed the best in terms of linguistic diversity, as shown in the table. Furthermore, GPT-4 excelled in maximizing linguistic variations, as indicated in the diversity vs. models plot in Fig. 11.

E Selecting the optimal paraphrase

E.1 Cosine Similarity

Cosine similarity is a metric used to measure the similarity between two vectors, often in high-dimensional spaces. It calculates the cosine of the angle between the two vectors, providing a numerical value that indicates how closely related they are.

In natural language processing, cosine similarity is often employed to assess the similarity between two documents represented as vectors in a high-dimensional space, where each dimension

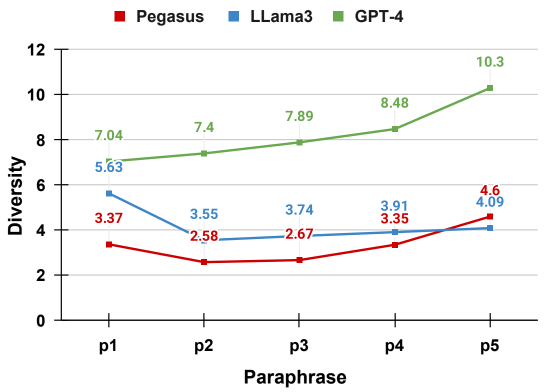


Figure 11: This figure shows the various parameters for generating paraphrases.

corresponds to a term or word. The cosine similarity ranges from -1 (entirely dissimilar) to 1 (completely similar), with 0 indicating orthogonality (no similarity).

The cosine similarity formula between vectors A and B is given in Eq. (1).

$$\text{Cosine Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \quad (1)$$

E.2 Topic Modeling

Topic modeling is a statistical technique for identifying topics in a collection of text documents. The goal is to uncover the hidden thematic structure within the text data. One standard algorithm used for topic modeling is Latent Dirichlet Allocation (LDA).

In topic modeling, each document in the corpus is considered a mixture of various topics, each represented as a distribution of words. The algorithm analyzes the co-occurrence patterns of words across documents to identify these latent topics. It helps understand the main themes or subjects present in a large collection of textual data without the need for manual annotation.

Topic modeling has applications in various NLP tasks, including document categorization, information retrieval, and content recommendation. It enables researchers and practitioners to gain insights into the underlying themes and structures within large textual datasets, making it a valuable tool for text analysis and understanding.

E.2.1 Topic Similarity

To overcome the issue of lengthy prompts, (Goyal et al., 2024) introduces the idea of inserting [PAUSE] tokens. However, it is not clear where these tokens can be added. Since they follow a rather random approach, we use a more deterministic approach in this work.

F Experimental Details

For different fine-tuning techniques, the list of hyperparameters is provided in Table 4.

Parameter	Value
FC1 size	768
FC2 size	600
Number of epochs	5
Learning rate	1E-03
Optimizer	AdamW
Dropout probability	0.1
Batch size	1

Table 4: Hyperparameters for different fine-tuning techniques.

G Factuality based entailment

This approach submits the prompt to the Google Search API to retrieve the top 20 relevant search results. From these 20 results, we assess a total of n sentences for their pertinence to the prompt using a similarity metric. The top 20 sentences most akin to the prompt are chosen. We utilize a textual entailment model to evaluate their credibility individually for each of the m sentences in the AI-generated text and the selected top 20 sentences. Based on the entailment scores, we classify the AI-generated text into three categories: (i) *support*, (ii) *refute*, and (iii) *not enough information*.

As far as we know, there is currently no SoTA method proposed for "automatic hallucination detection". There are other associated challenges: With new LLMs being released weekly, there is an urgent need to enhance automatic hallucination detection and mitigation techniques. While using a benchmark is currently standard practice in the NLP community, the rapid pace of change necessitates a deeper understanding of how newer LLMs induce hallucinations. Strict adherence to a fixed benchmark, released a year (let's say) ago, risks overlooking advancements in the field due to the rapid pace of development. Let's consider the HILT paper as the current SoTA in hallucination mitigation techniques for discussion. Our study focuses on 22 LLMs. Consequently, the challenge arises: how can we assess the efficacy of our proposed mitigation techniques for these newer models when no SoTA dataset is available for them? Now, let's delve into the challenges associated with automatically evaluating hallucination mitigation. The HILT dataset comprises prompts, LLM-generated text, and annotated sentences identified as hallucinated. However, no reference data points indicate what would have been a factually correct generation in place of those hallucinated sentences. To our knowledge, no other dataset containing such information exists. On another note, suppose we or other researchers propose a technique for hallucination mitigation. How can we ascertain whether the newer generations, after incorporating these proposed techniques, exhibit reduced or eliminated hallucinations? Without a benchmark or baseline to compare against, it is currently infeasible to automatically assess the effectiveness of hallucination mitigation techniques. Let's assume we possess a dataset that includes the crucial component missing in previous studies: what would constitute a factually correct generation given a specific prompt? From existing research, we understand that LLMs are highly sensitive to even minor prompt alterations. Consequently, LLM-generated outputs may deviate significantly from human annotations. As a result, it may be necessary to have multiple annotations and utilize metrics such as BLEU, ROUGE, and BERTScore to gauge similarity. However, these metrics may or may not effectively capture factual correctness. The evaluation of the factual accuracy of LLM outputs necessitates the development of a reliable method and metric, which, regrettably, has yet to be proposed. We propose an alternative hallucination mitigation evaluation approach: employing an overall entailment-based method to evaluate the extent to which retrieved facts support LLM generations. This methodology is straightforward to implement and can be scaled effectively. We aim to assess whether newer proposed mitigation methods enhance overall entailment support. While this approach may be indirect, we believe it is the most feasible option given the limitations discussed earlier. Without it, conducting experiments on the scale of 22 LLMs and a dataset of 90k samples would be exceedingly tricky.

H Results after adding `[PAUSE]` tokens

In the Table 5 below, we show the experimental results for adding `[PAUSE]`.

I Selecting the optimal paraphrased prompt

The detailed explanation of our algorithm to identify the optimal paraphrased prompt is provided in the illustration in Fig. 12.

J Before and after adding `[PAUSE]` token

In the Figs. 13 to 23 below, we demonstrate how adding a `[PAUSE]` token affects the comprehension of longer prompts across a subset of selected LLMs.

Fine-tuning technique	Person			Location			Numeric			Time		
	Support	Refute	Neutral	Support	Refute	Neutral	Support	Refute	Neutral	Support	Refute	Neutral
Original Prompt	0.63	0.54	0.78	0.52	0.55	0.77	0.22	0.89	0.77	0.29	0.65	0.72
Optimal Paraphrase + LDA topics	0.65	0.26	0.59	0.59	0.28	0.54	0.36	0.36	0.66	0.44	0.56	0.7
+ [PAUSE] Injection												
Optimal Paraphrase + LDA topics + w/ [PAUSE] token LoRA	0.7	0.19	0.69	0.61	0.25	0.53	0.53	0.29	0.69	0.59	0.29	0.72
Optimal Paraphrase + LDA topics + w/ [PAUSE] token QALoRA	0.72	0.21	0.67	0.62	0.22	0.52	0.58	0.32	0.67	0.62	0.31	0.73
Optimal Paraphrase + LDA topics + w/ [PAUSE] token Reverse Knowledge Distillation	0.86	0.12	0.79	0.77	0.18	0.48	0.69	0.26	0.79	0.68	0.23	0.66

Table 5: Empirical results for Reverse Knowledge Distillation with [PAUSE] tokens.

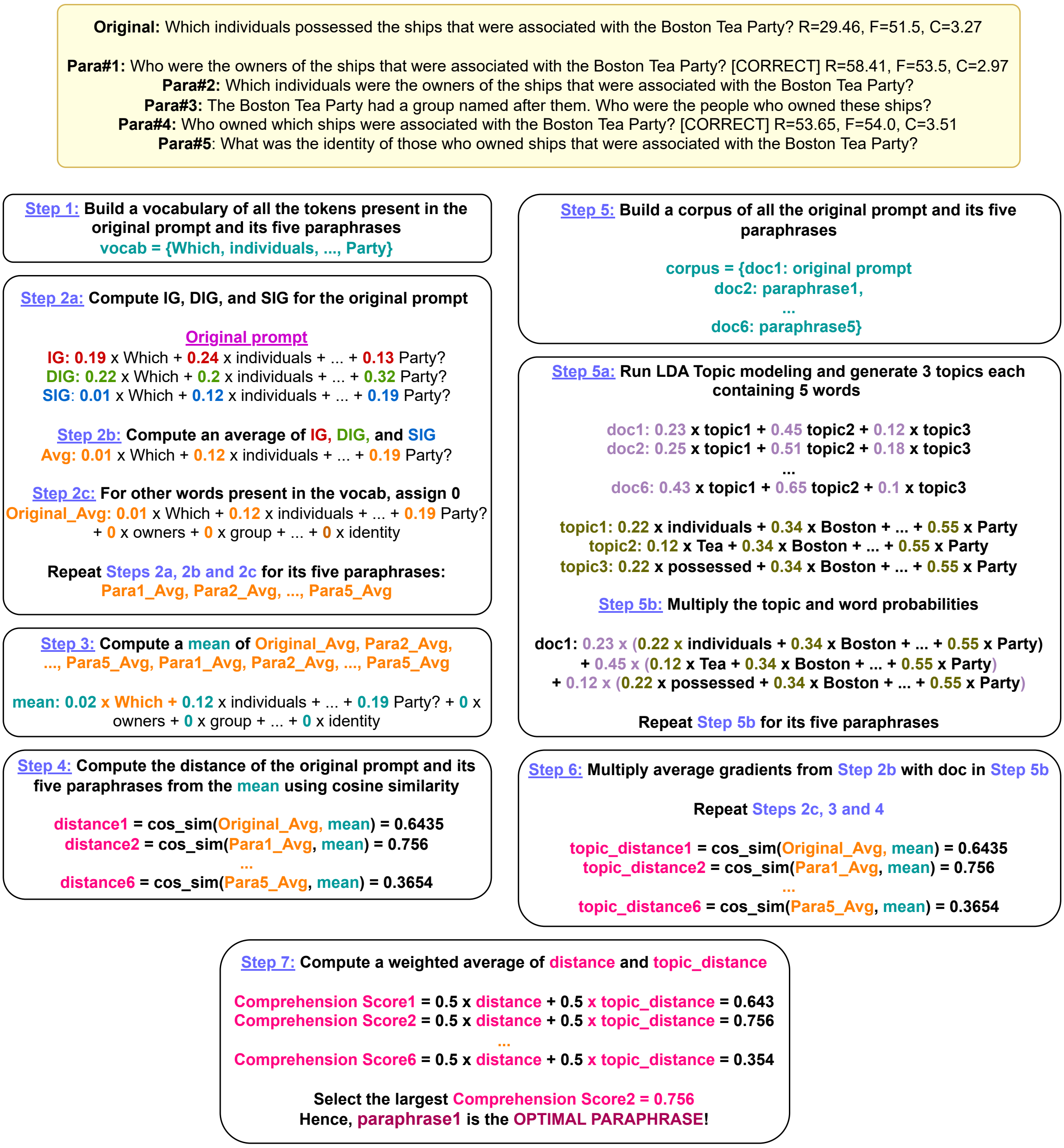
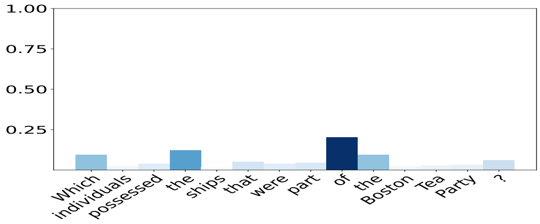
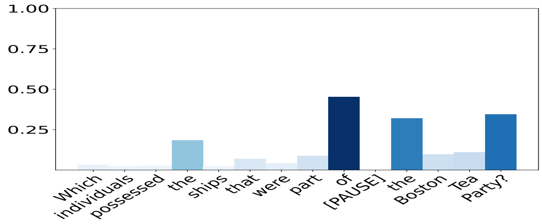


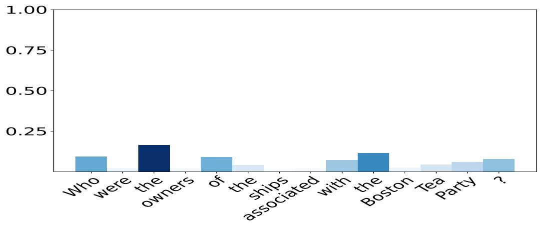
Figure 12: A walkthrough of our optimal paraphrase selection process.



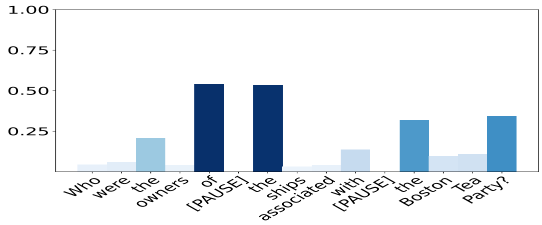
(a) Before adding [PAUSE] tokens to original prompt.



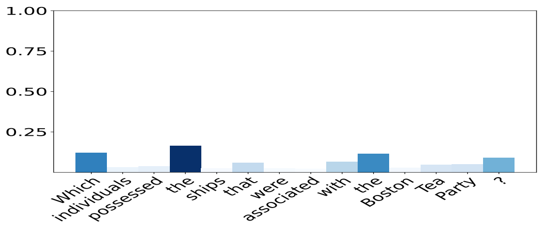
(b) After adding [PAUSE] tokens to original prompt.



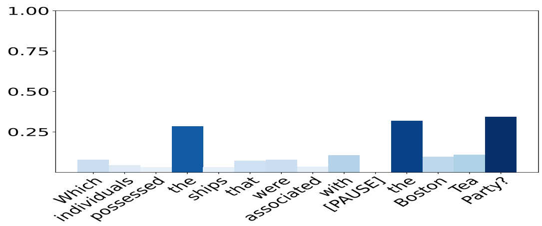
(c) Before adding [PAUSE] tokens to paraphrase 1.



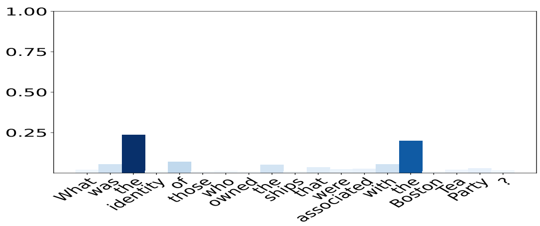
(d) After adding [PAUSE] tokens to paraphrase 1.



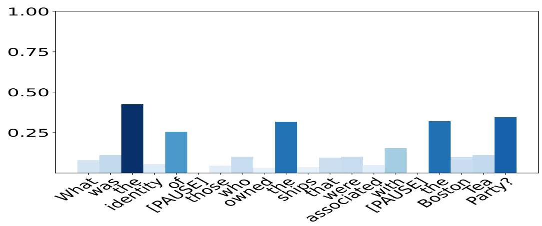
(e) Before adding [PAUSE] tokens to paraphrase 2.



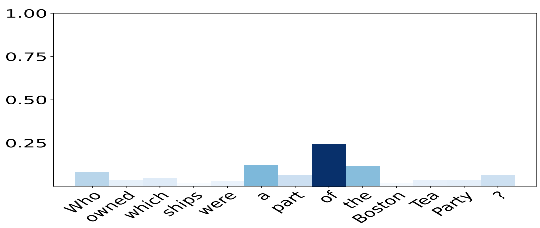
(f) After adding [PAUSE] tokens to paraphrase 2.



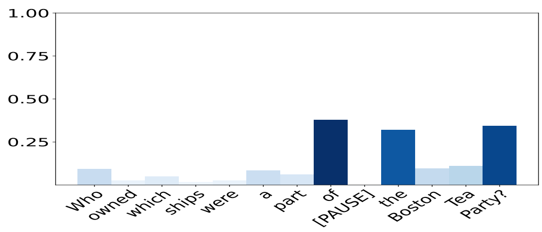
(g) Before adding [PAUSE] tokens to paraphrase 3.



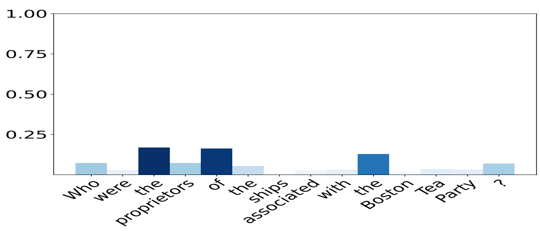
(h) After adding [PAUSE] tokens to paraphrase 3.



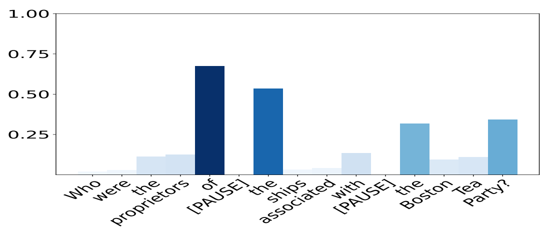
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

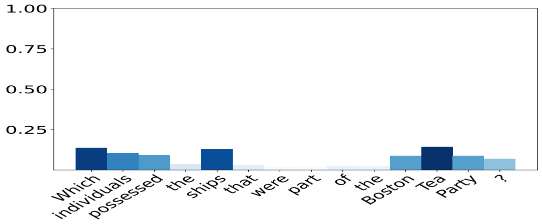


(k) Before adding [PAUSE] tokens to paraphrase 5.

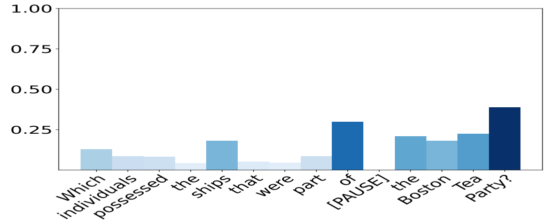


(l) After adding [PAUSE] tokens to paraphrase 5.

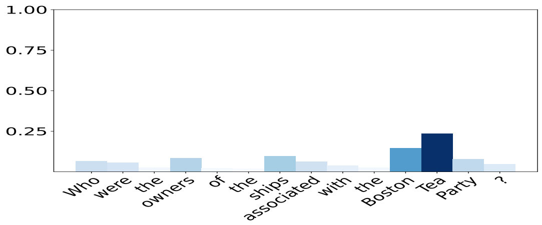
Figure 13: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for alpaca.



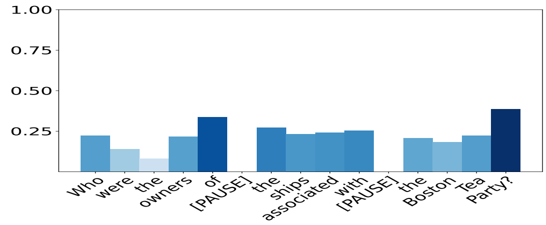
(a) Before adding [PAUSE] tokens to original prompt.



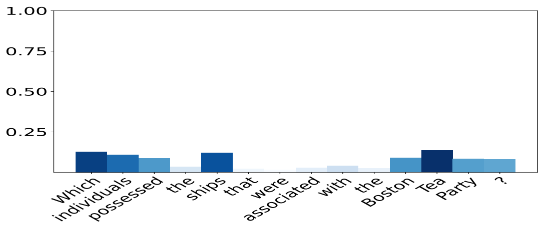
(b) After adding [PAUSE] tokens to original prompt.



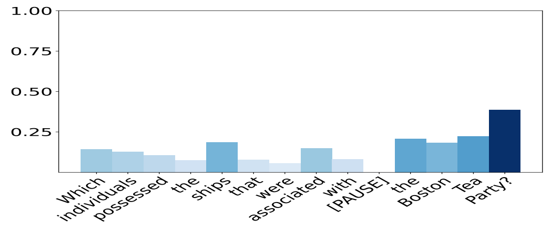
(c) Before adding [PAUSE] tokens to paraphrase 1.



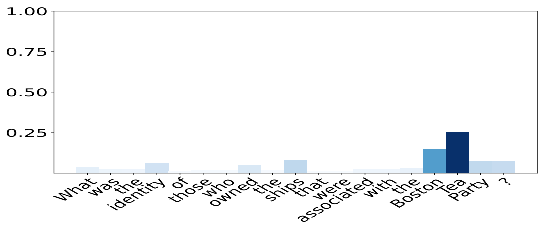
(d) After adding [PAUSE] tokens to paraphrase 1.



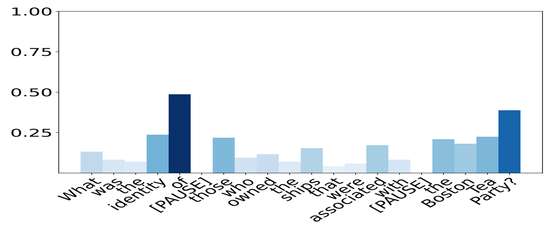
(e) Before adding [PAUSE] tokens to paraphrase 2.



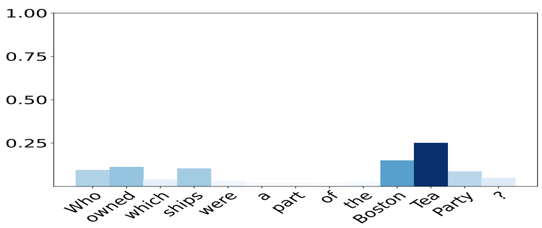
(f) After adding [PAUSE] tokens to paraphrase 2.



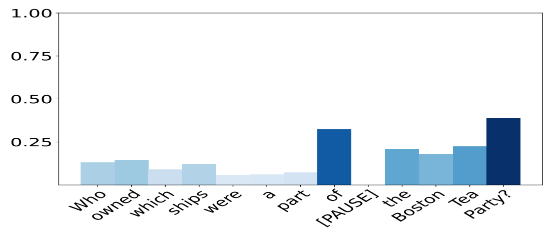
(g) Before adding [PAUSE] tokens to paraphrase 3.



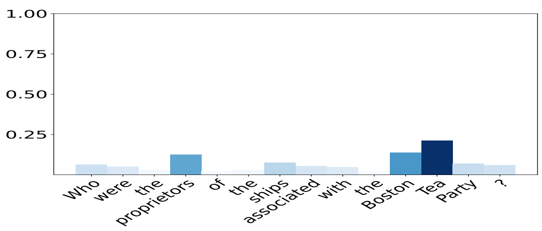
(h) After adding [PAUSE] tokens to paraphrase 3.



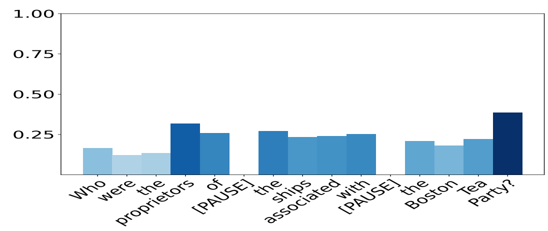
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

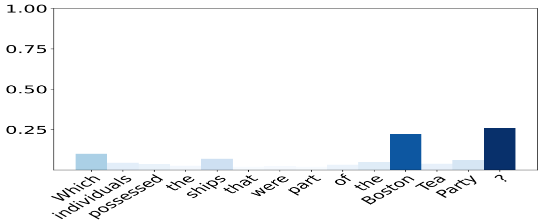


(k) Before adding [PAUSE] tokens to paraphrase 5.

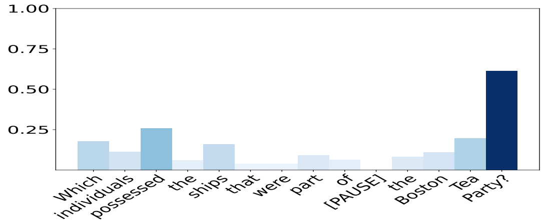


(l) After adding [PAUSE] tokens to paraphrase 5.

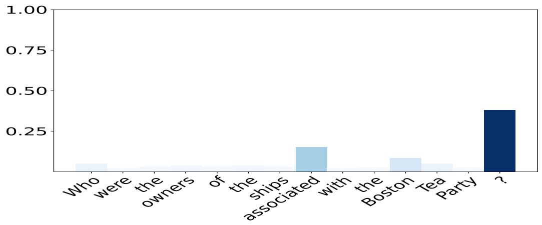
Figure 14: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for bloomz.



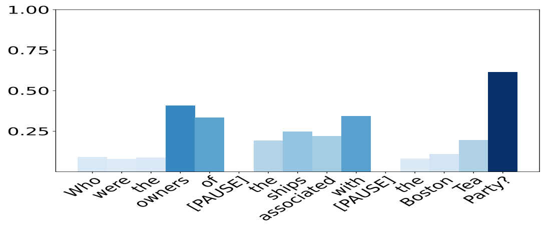
(a) Before adding [PAUSE] tokens to original prompt.



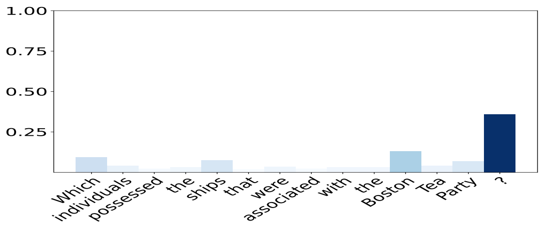
(b) After adding [PAUSE] tokens to original prompt.



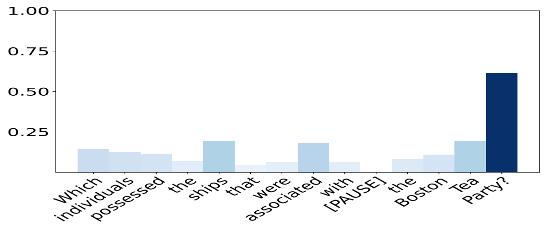
(c) Before adding [PAUSE] tokens to paraphrase 1.



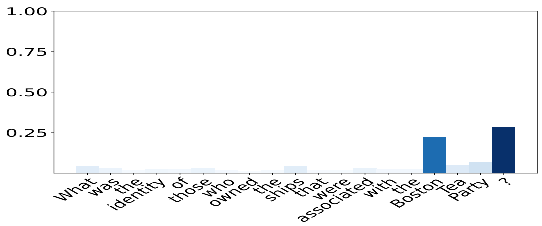
(d) After adding [PAUSE] tokens to paraphrase 1.



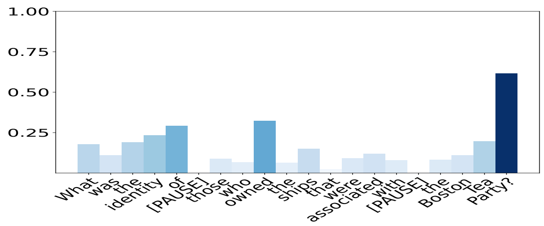
(e) Before adding [PAUSE] tokens to paraphrase 2.



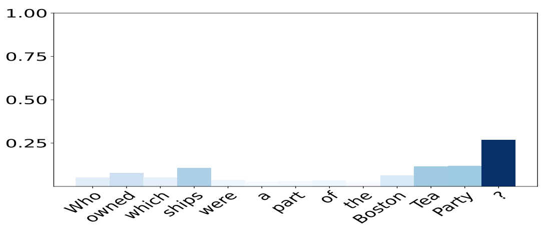
(f) After adding [PAUSE] tokens to paraphrase 2.



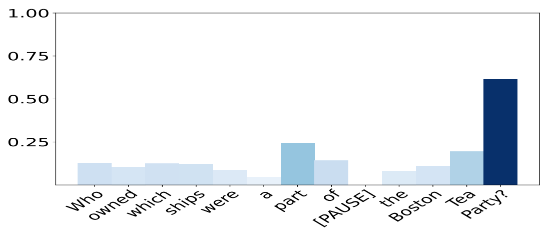
(g) Before adding [PAUSE] tokens to paraphrase 3.



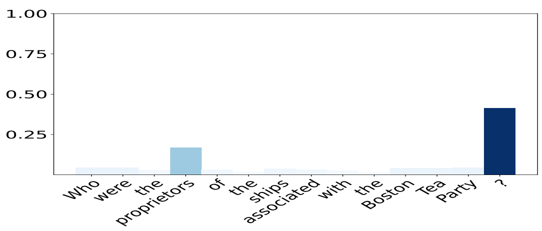
(h) After adding [PAUSE] tokens to paraphrase 3.



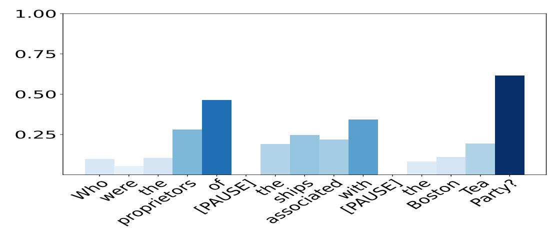
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

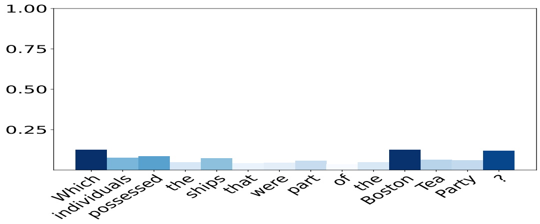


(k) Before adding [PAUSE] tokens to paraphrase 5.

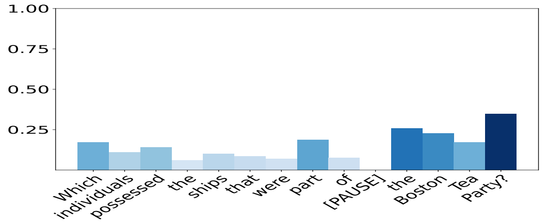


(l) After adding [PAUSE] tokens to paraphrase 5.

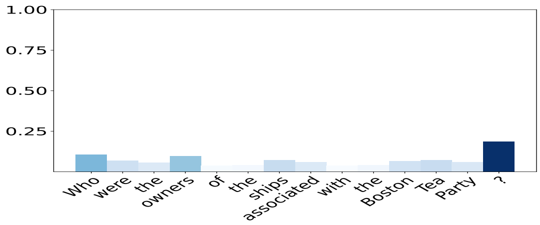
Figure 15: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for dolly.



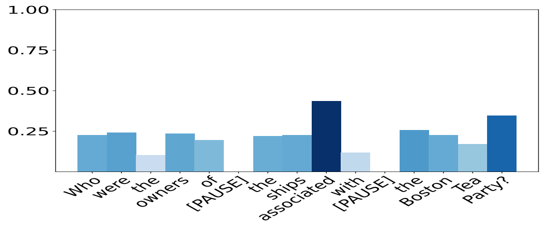
(a) Before adding [PAUSE] tokens to original prompt.



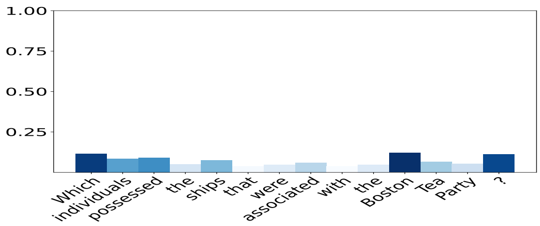
(b) After adding [PAUSE] tokens to original prompt.



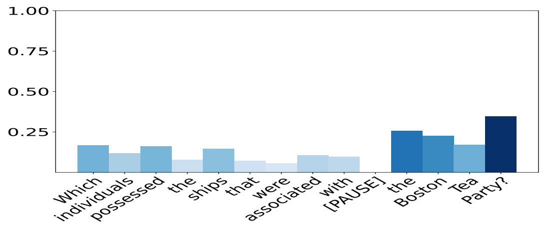
(c) Before adding [PAUSE] tokens to paraphrase 1.



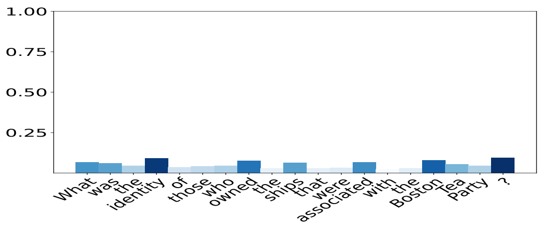
(d) After adding [PAUSE] tokens to paraphrase 1.



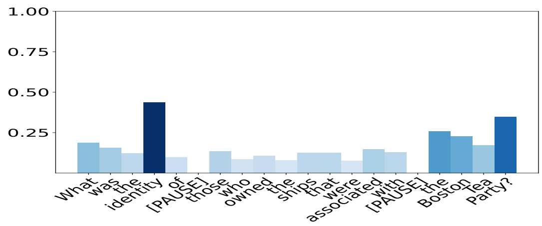
(e) Before adding [PAUSE] tokens to paraphrase 2.



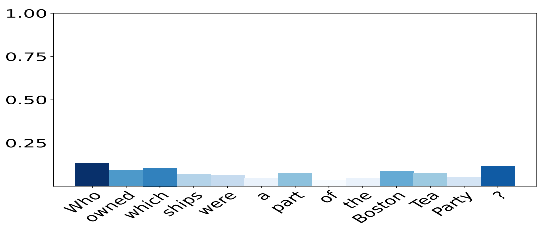
(f) After adding [PAUSE] tokens to paraphrase 2.



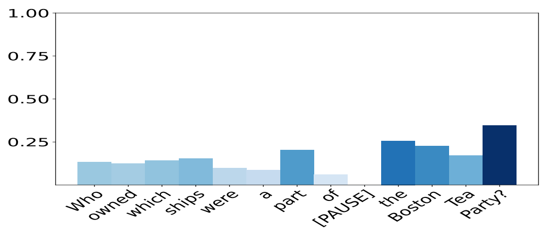
(g) Before adding [PAUSE] tokens to paraphrase 3.



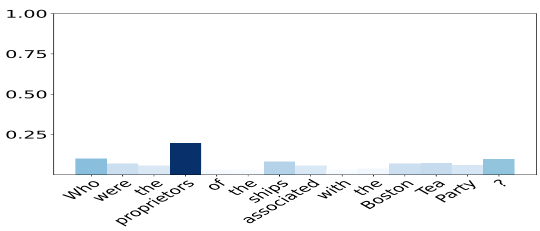
(h) After adding [PAUSE] tokens to paraphrase 3.



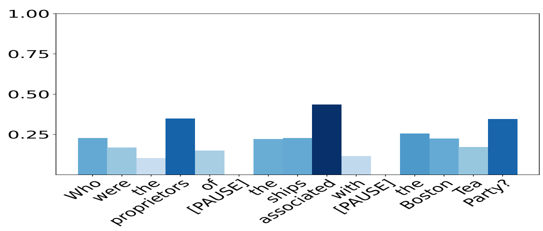
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

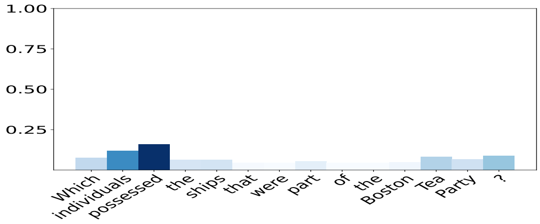


(k) Before adding [PAUSE] tokens to paraphrase 5.

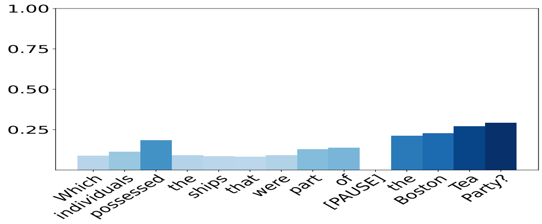


(l) After adding [PAUSE] tokens to paraphrase 5.

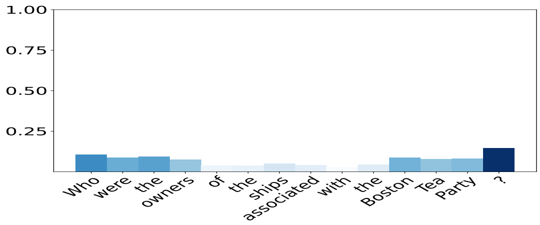
Figure 16: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for Falcon.



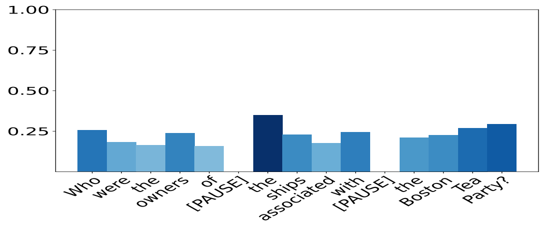
(a) Before adding [PAUSE] tokens to original prompt.



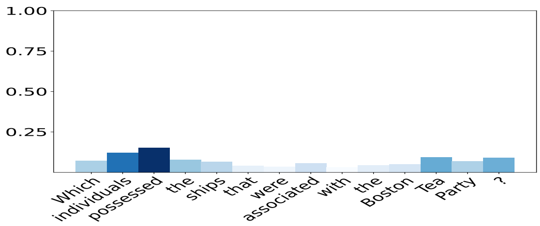
(b) After adding [PAUSE] tokens to original prompt.



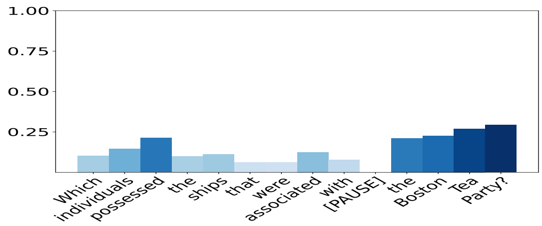
(c) Before adding [PAUSE] tokens to paraphrase 1.



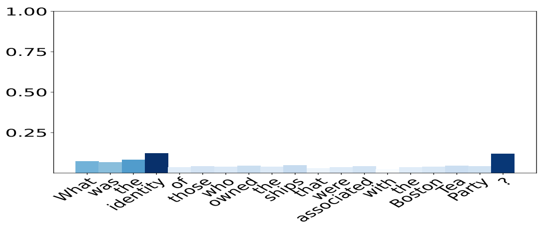
(d) After adding [PAUSE] tokens to paraphrase 1.



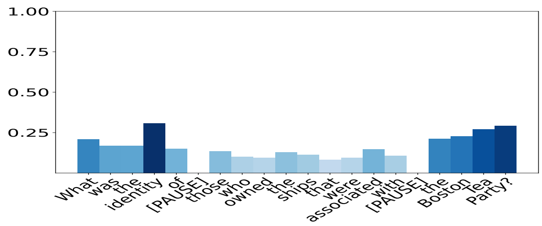
(e) Before adding [PAUSE] tokens to paraphrase 2.



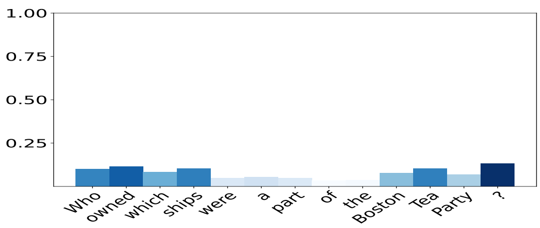
(f) After adding [PAUSE] tokens to paraphrase 2.



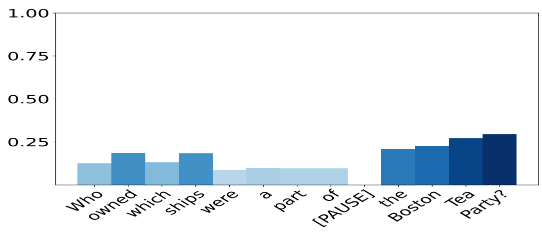
(g) Before adding [PAUSE] tokens to paraphrase 3.



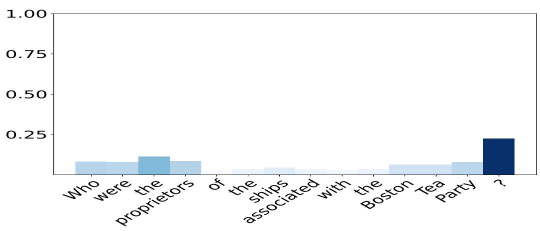
(h) After adding [PAUSE] tokens to paraphrase 3.



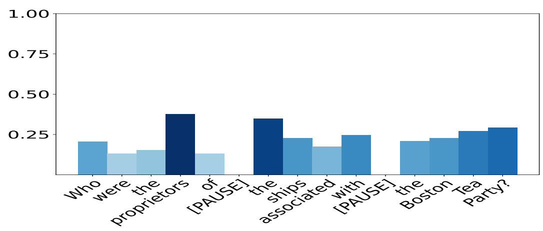
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

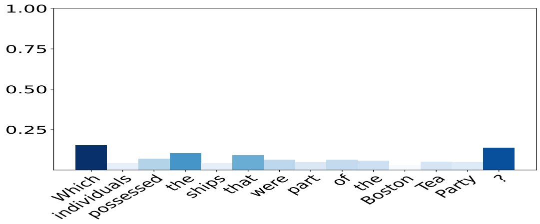


(k) Before adding [PAUSE] tokens to paraphrase 5.

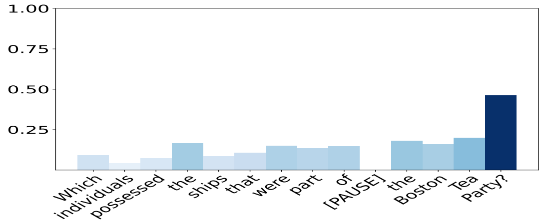


(l) After adding [PAUSE] tokens to paraphrase 5.

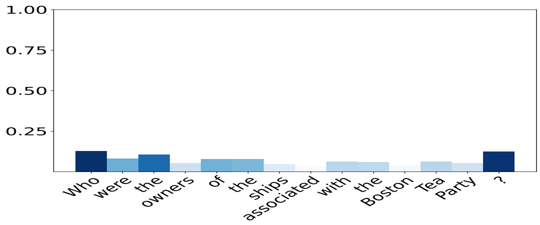
Figure 17: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for FLAN-T5.



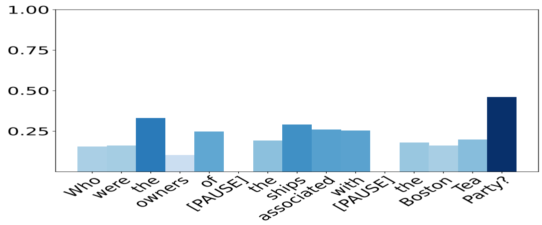
(a) Before adding [PAUSE] tokens to original prompt.



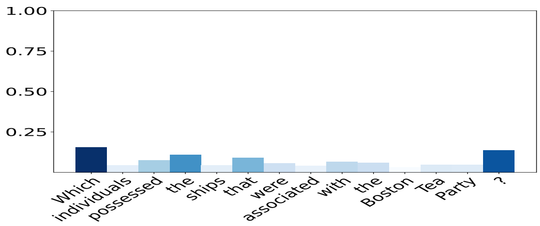
(b) After adding [PAUSE] tokens to original prompt.



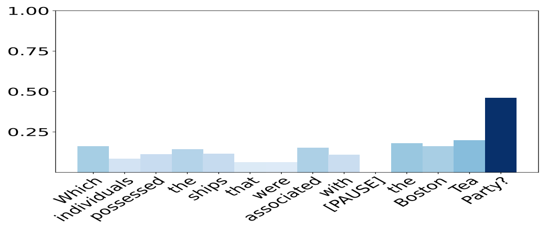
(c) Before adding [PAUSE] tokens to paraphrase 1.



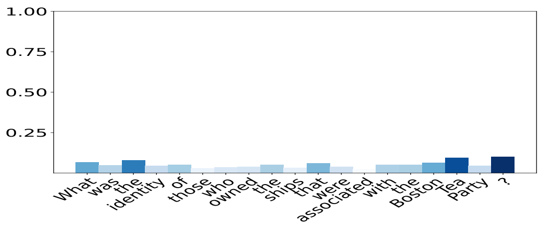
(d) After adding [PAUSE] tokens to paraphrase 1.



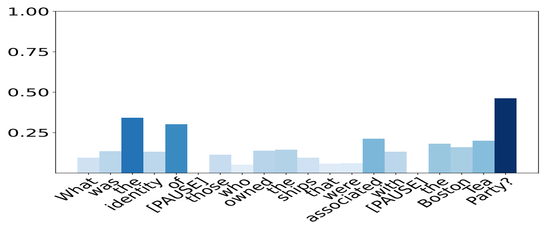
(e) Before adding [PAUSE] tokens to paraphrase 2.



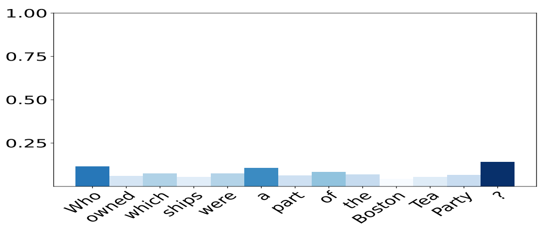
(f) After adding [PAUSE] tokens to paraphrase 2.



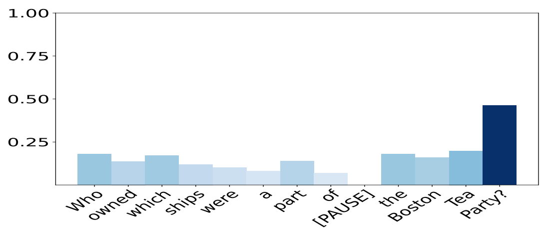
(g) Before adding [PAUSE] tokens to paraphrase 3.



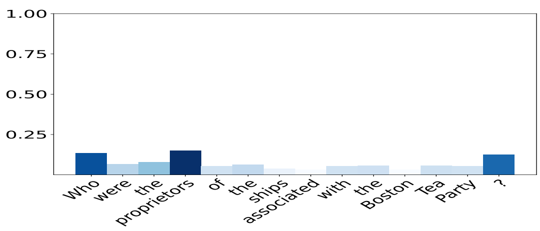
(h) After adding [PAUSE] tokens to paraphrase 3.



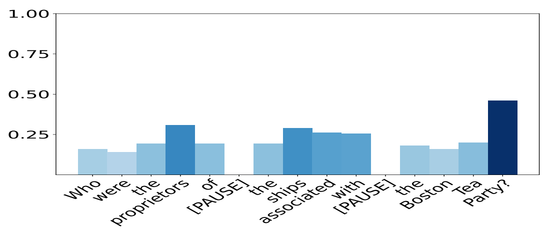
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

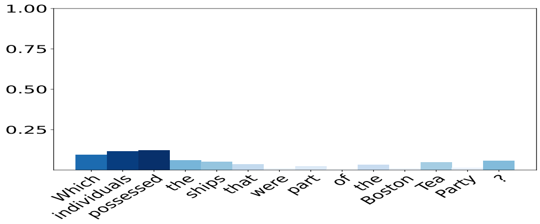


(k) Before adding [PAUSE] tokens to paraphrase 5.

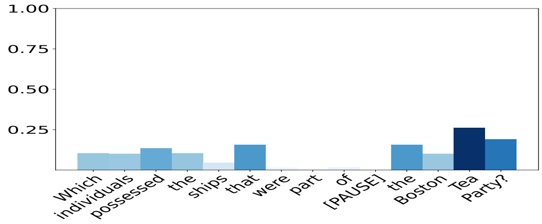


(l) After adding [PAUSE] tokens to paraphrase 5.

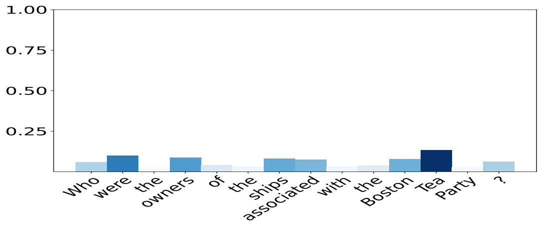
Figure 18: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for GPT Neo.



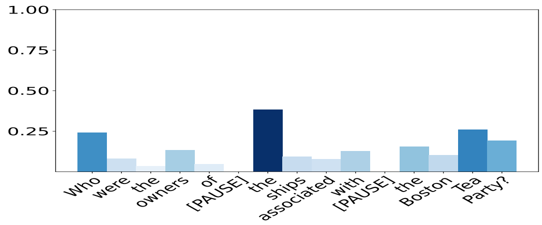
(a) Before adding [PAUSE] tokens to original prompt.



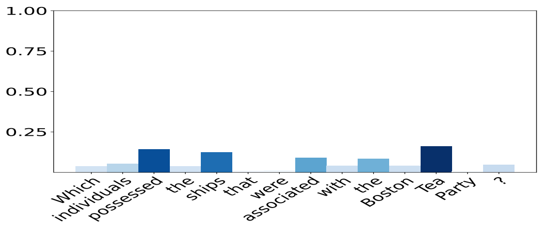
(b) After adding [PAUSE] tokens to original prompt.



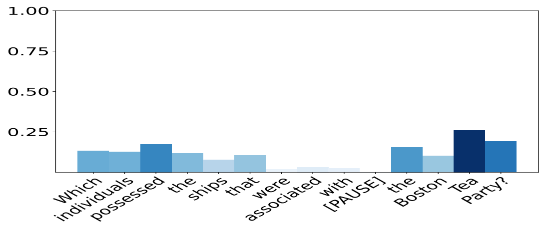
(c) Before adding [PAUSE] tokens to paraphrase 1.



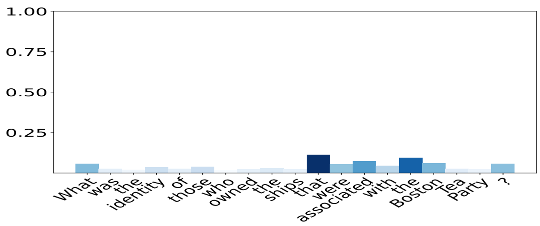
(d) After adding [PAUSE] tokens to paraphrase 1.



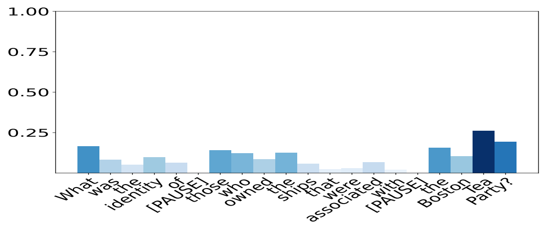
(e) Before adding [PAUSE] tokens to paraphrase 2.



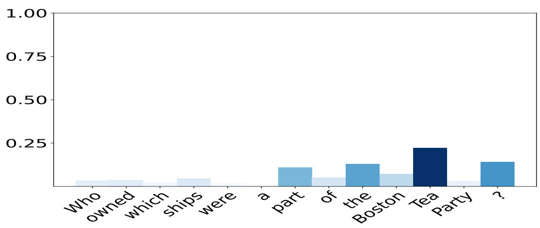
(f) After adding [PAUSE] tokens to paraphrase 2.



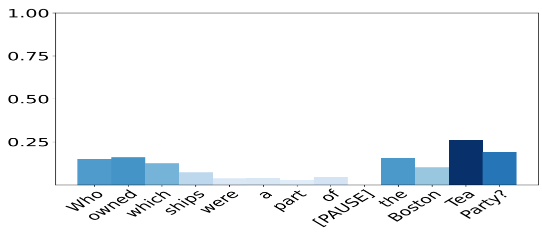
(g) Before adding [PAUSE] tokens to paraphrase 3.



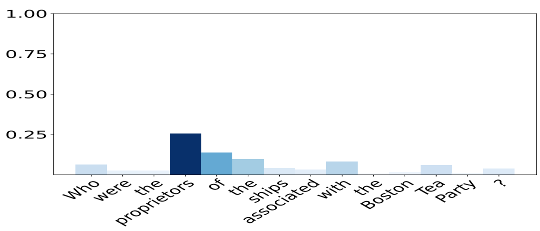
(h) After adding [PAUSE] tokens to paraphrase 3.



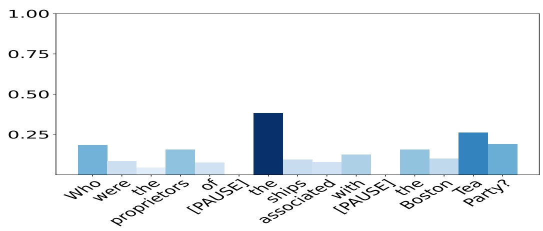
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

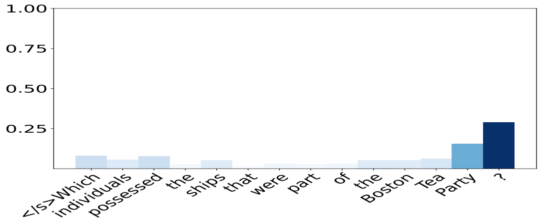


(k) Before adding [PAUSE] tokens to paraphrase 5.

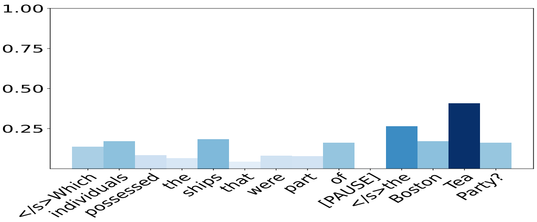


(l) After adding [PAUSE] tokens to paraphrase 5.

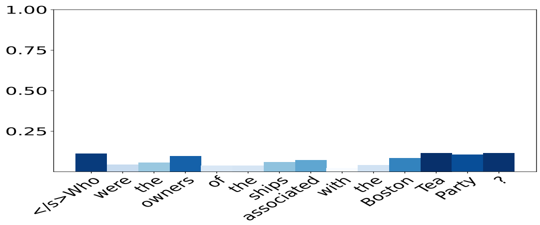
Figure 19: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for Llama2.



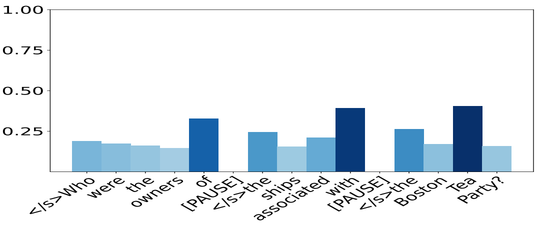
(a) Before adding [PAUSE] tokens to original prompt.



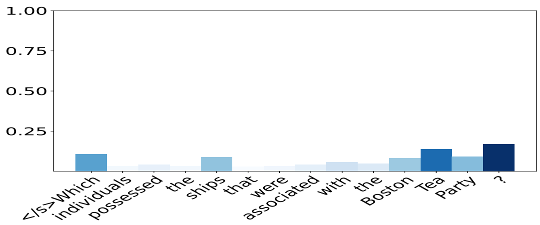
(b) After adding [PAUSE] tokens to original prompt.



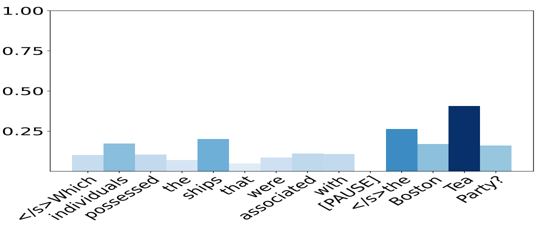
(c) Before adding [PAUSE] tokens to paraphrase 1.



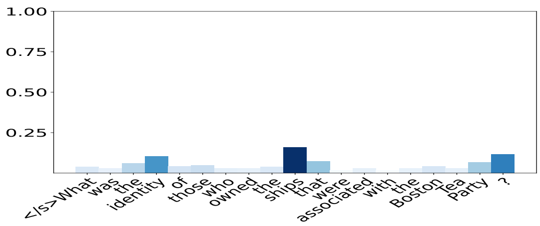
(d) After adding [PAUSE] tokens to paraphrase 1.



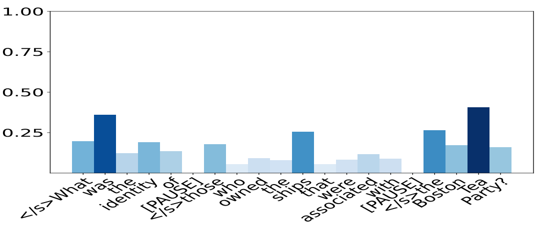
(e) Before adding [PAUSE] tokens to paraphrase 2.



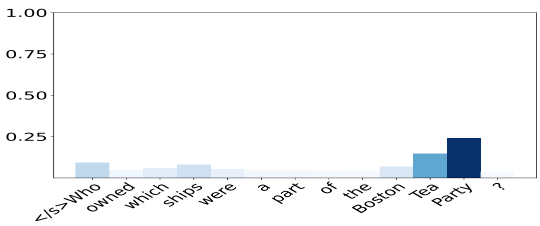
(f) After adding [PAUSE] tokens to paraphrase 2.



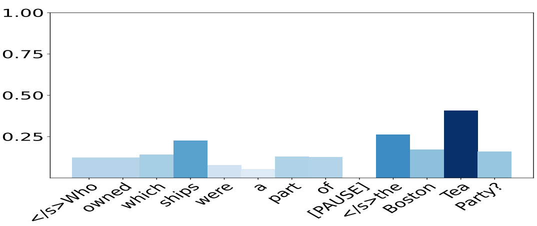
(g) Before adding [PAUSE] tokens to paraphrase 3.



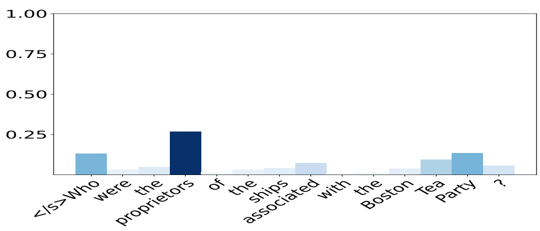
(h) After adding [PAUSE] tokens to paraphrase 3.



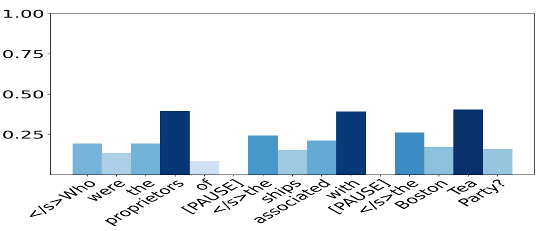
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

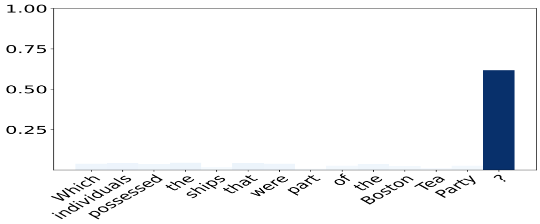


(k) Before adding [PAUSE] tokens to paraphrase 5.

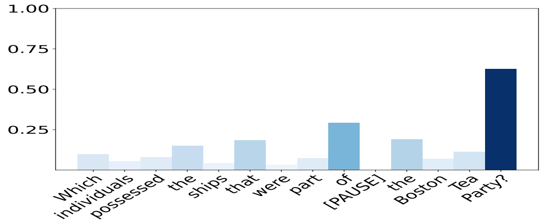


(l) After adding [PAUSE] tokens to paraphrase 5.

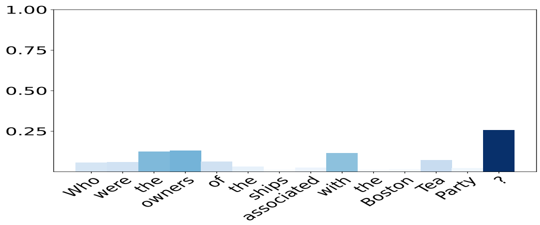
Figure 20: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for OPT.



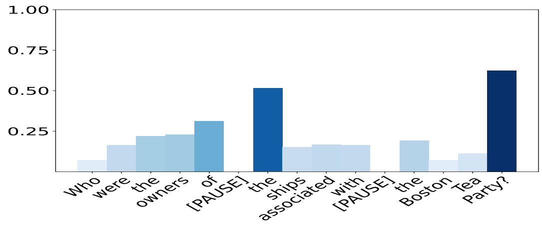
(a) Before adding [PAUSE] tokens to original prompt.



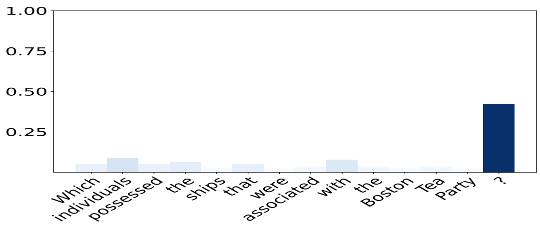
(b) After adding [PAUSE] tokens to original prompt.



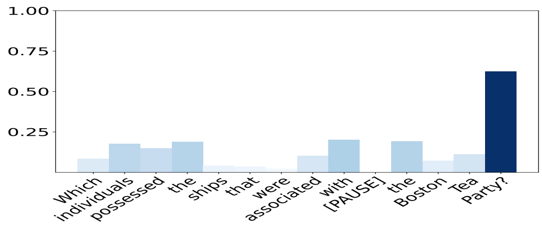
(c) Before adding [PAUSE] tokens to paraphrase 1.



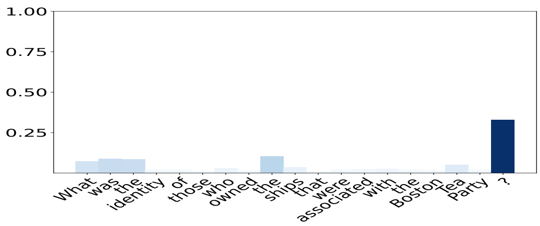
(d) After adding [PAUSE] tokens to paraphrase 1.



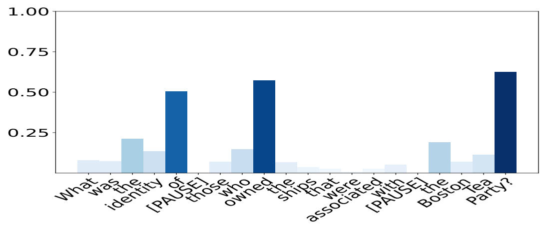
(e) Before adding [PAUSE] tokens to paraphrase 2.



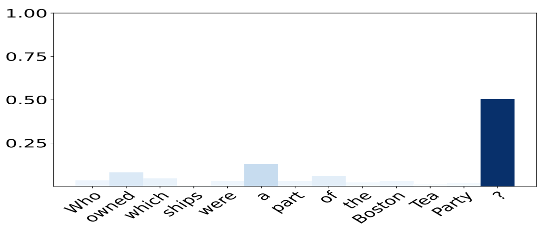
(f) After adding [PAUSE] tokens to paraphrase 2.



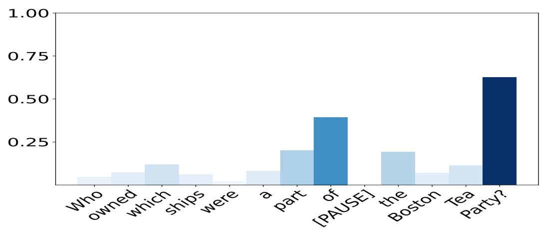
(g) Before adding [PAUSE] tokens to paraphrase 3.



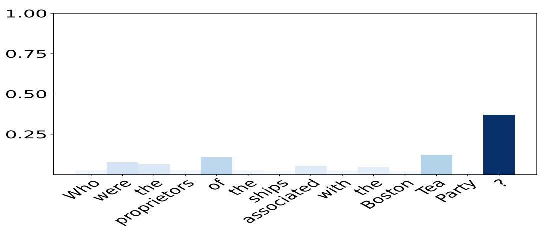
(h) After adding [PAUSE] tokens to paraphrase 3.



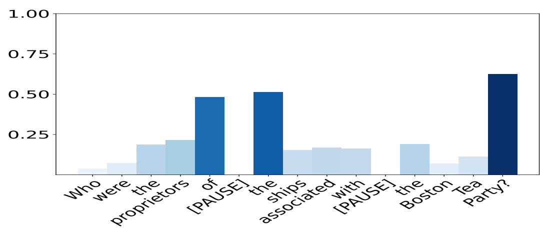
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

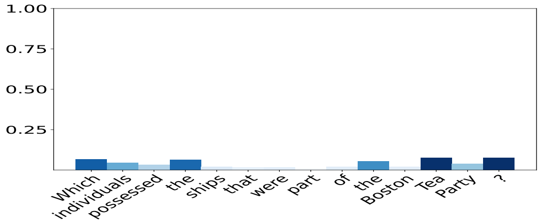


(k) Before adding [PAUSE] tokens to paraphrase 5.

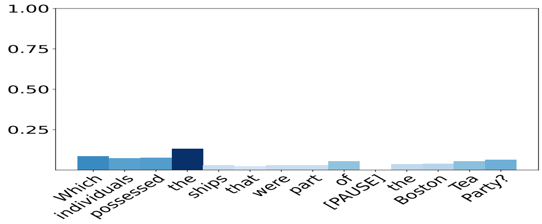


(l) After adding [PAUSE] tokens to paraphrase 5.

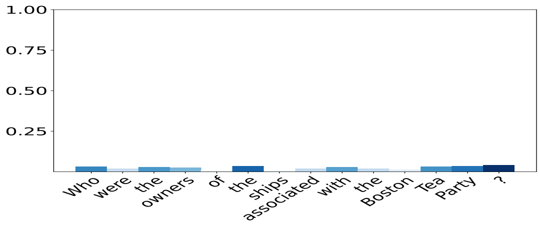
Figure 21: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for phi-2.



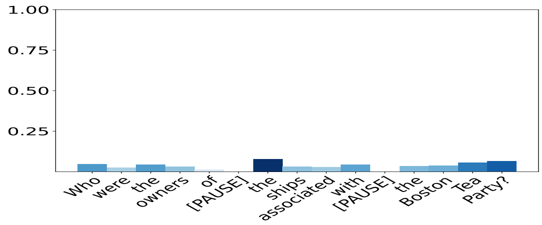
(a) Before adding [PAUSE] tokens to original prompt.



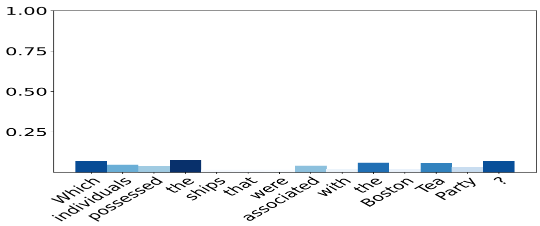
(b) After adding [PAUSE] tokens to original prompt.



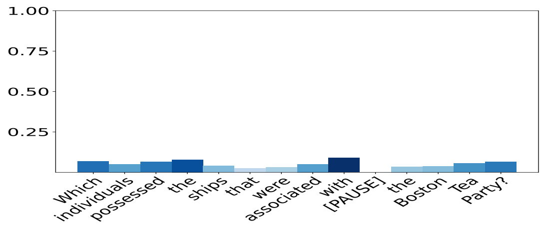
(c) Before adding [PAUSE] tokens to paraphrase 1.



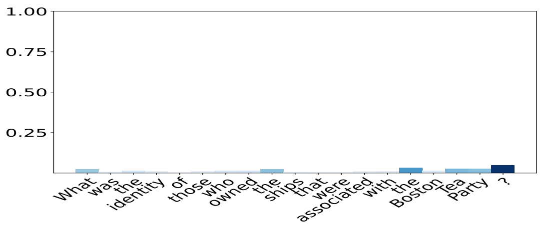
(d) After adding [PAUSE] tokens to paraphrase 1.



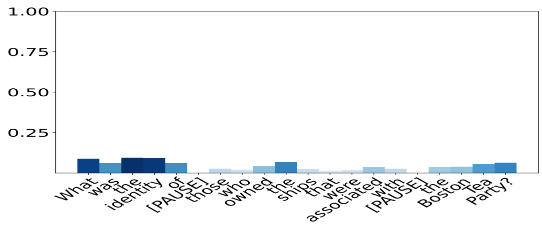
(e) Before adding [PAUSE] tokens to paraphrase 2.



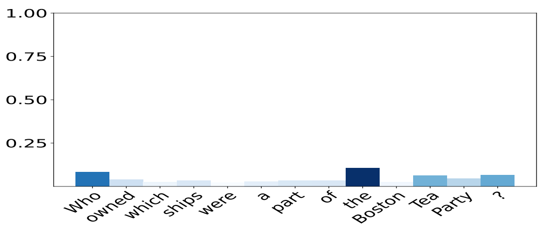
(f) After adding [PAUSE] tokens to paraphrase 2.



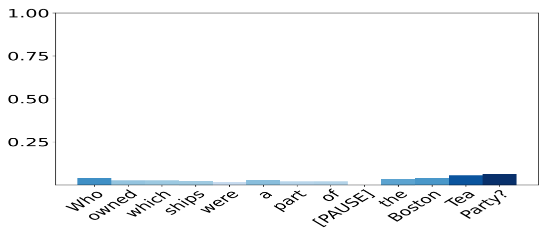
(g) Before adding [PAUSE] tokens to paraphrase 3.



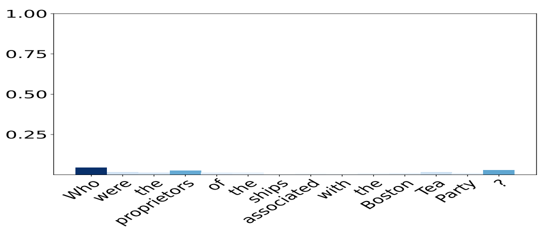
(h) After adding [PAUSE] tokens to paraphrase 3.



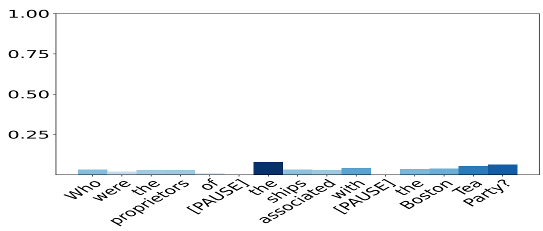
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.

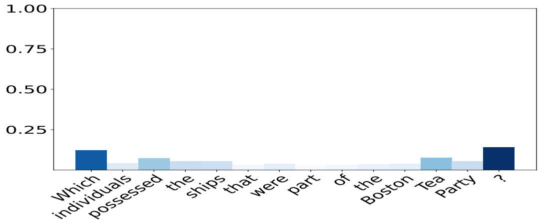


(k) Before adding [PAUSE] tokens to paraphrase 5.

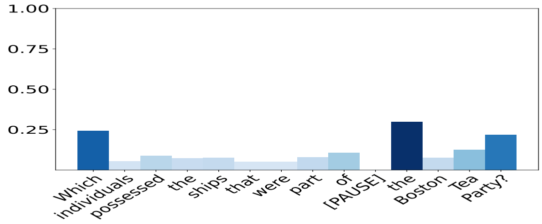


(l) After adding [PAUSE] tokens to paraphrase 5.

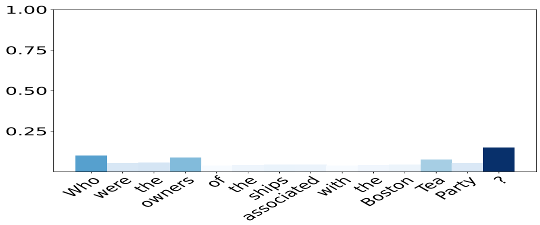
Figure 22: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for Vicuna.



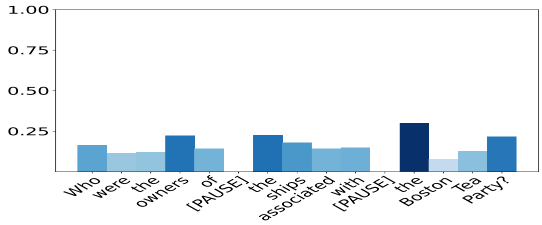
(a) Before adding [PAUSE] tokens to original prompt.



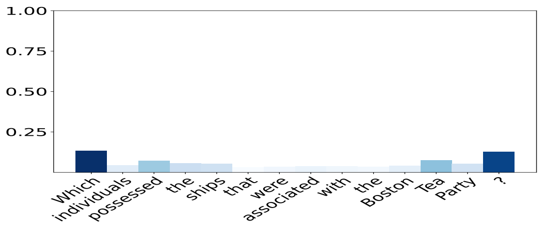
(b) After adding [PAUSE] tokens to original prompt.



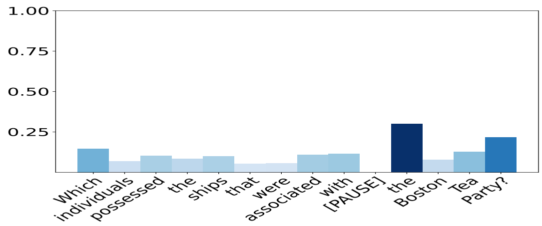
(c) Before adding [PAUSE] tokens to paraphrase 1.



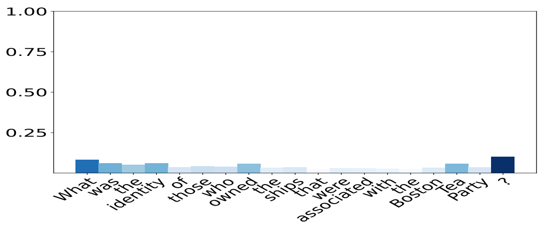
(d) After adding [PAUSE] tokens to paraphrase 1.



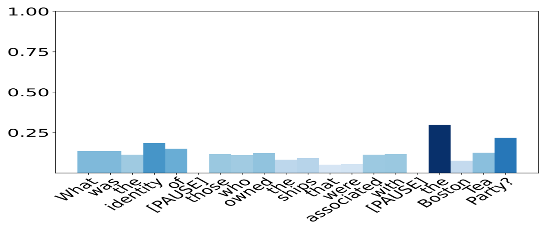
(e) Before adding [PAUSE] tokens to paraphrase 2.



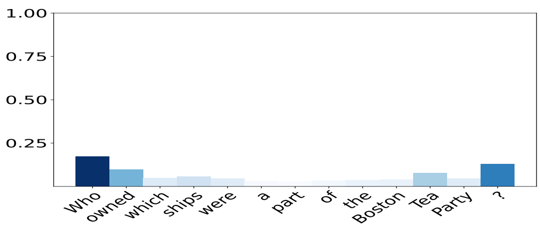
(f) After adding [PAUSE] tokens to paraphrase 2.



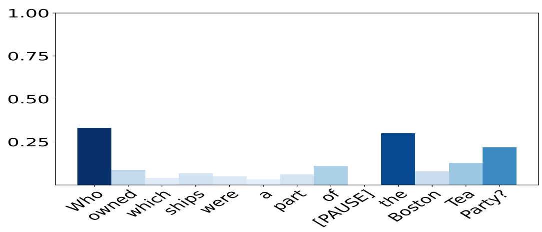
(g) Before adding [PAUSE] tokens to paraphrase 3.



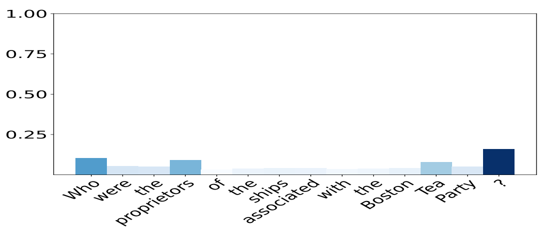
(h) After adding [PAUSE] tokens to paraphrase 3.



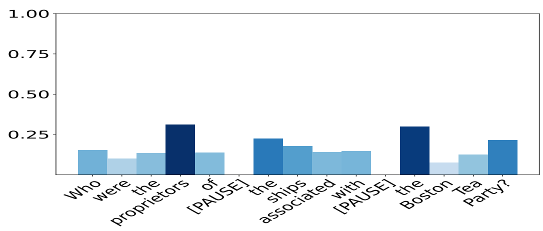
(i) Before adding [PAUSE] tokens to paraphrase 4.



(j) After adding [PAUSE] tokens to paraphrase 4.



(k) Before adding [PAUSE] tokens to paraphrase 5.



(l) After adding [PAUSE] tokens to paraphrase 5.

Figure 23: The phrase **Boston Tea** gets more importance score after adding [PAUSE] token for Zephyr.