Improving the Faithfulness of LLM-based Abstractive Summarization with Span-level Unlikelihood Training

Sicong Huang **Qianqi Yan** **Shengze Wang** **Ian Lane**
University of California, Santa Cruz
{shuan213, qyan79, shengze, ialane}@ucsc.edu

Abstract

Abstractive summarization using large language models (LLMs) has become an essential tool for condensing information. Despite their ability to generate fluent summaries, these models often produce texts that are unfaithful to the original documents, manifested through hallucinations of specific words, phrases, or concepts. Current approaches to mitigating unfaithfulness typically involve post-processing corrections or contrastive learning from synthetically generated negative samples, which do not fully address the spectrum of errors that can arise in LLM-generated summaries. In this paper, we introduce a novel approach to fine-tune LLMs specifically to reduce the occurrence of unfaithful spans of text in generated summaries. We first annotate span-level hallucinations in LLM-generated summaries using automatic labeling with GPT-4. We then fine-tune the LLM using both summaries with no hallucinations and spans of hallucinated text to improve the faithfulness of the model. This paper introduces a dataset labeled to distinguish between faithful and unfaithful content and compare the performance of three techniques: *gradient ascent*, *unlikelihood training*, and *task vector negation*. Our experimental results show that unlikelihood training can effectively use span-level annotations to enhance summary faithfulness, reducing the number of summaries with hallucinations from 31% to 13%, a reduction of 58% on the CNN summarization dataset and from 33% to 20%, a reduction of 39% on the SAMSum dataset.

1 Introduction

Abstractive summarization aims to condense text into a shorter version by distilling the key information from the source text and rewriting it in a concise manner. Recent advances in large language models (LLMs) such as GPT-4 (OpenAI et al., 2024) Llama 2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023) have significantly enhanced the capability of summarization systems to produce fluent and coherent summaries. Additionally, the growing adoption of retrieval-augmented generation (RAG) (Lewis et al., 2020; Siriwardhana et al., 2023; Zhang et al., 2024) has underscored the role of summarization as a critical component of modern interactive natural language systems.

However, despite the strong capabilities of LLMs, they still suffer from the problem of hallucination (Huang et al., 2023; Ji et al., 2023; Jiang et al., 2024), often referred to as unfaithfulness in summarization (Maynez et al., 2020; Goyal and Durrett, 2021; Kryscinski et al., 2020). This issue arises when the generated summary contains information that is neither grounded in nor aligned with the source document, which limits the practicality of deploying summarization systems in real applications. Figure 1 shows an example SAMSum dialogue and its corresponding generated summary that contains a span of hallucinated text.

A number of approaches have attempted to alleviate unfaithfulness with post-processing. For instance, Dong et al. (2020) and Cao et al. (2020) have suggested methods to edit and correct factual inaccuracies in summaries post-generation. Similarly, Madaan et al. (2023); Akyurek et al. (2023) employ the critique-and-refine process that generates critical feedback on the initial summary as a guide for the summarizer to refine the summary. Although effective, the extra post-processing steps required induce high latency and increase the computational demands during inference, restricting their applicable use cases.

Another option is to learn from negative samples. Cao and Wang (2021); Tang et al. (2022); Zhang et al. (2023) synthetically create negative samples of unfaithful summaries. These samples are derived from reference summaries using strategies that mimic common error types. However, there are three problems with this approach:

1. Human reviewers generally prefer LLM sum-

Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026), pages 425–438

Source:

Pam:	Hey Robert, you said you could help with Tom's birthday?
Robert:	Sure, what do you need?
Pam:	I have to go shopping, cook, and clean, and I figured out I don't have time to pick up the balloons.
Robert:	From where?
Pam:	There's this store in the city centre that sells these awesome floating balloons.
Robert:	No problem, just text me the address.
Pam:	Bless you!
Robert:	;)

Summary:

Pam asked Robert for help with Tom's birthday celebration, as she needs to go shopping, cook, and clean, and doesn't have time to pick up floating balloons from a store in the city centre. Robert agreed to help by providing the address of the store .

Figure 1: An example SAMSum dialogue and its corresponding model-generated summary that contains a hallucination span shown as highlighted text.

maries over standard reference summaries, even for large and well-established summarization datasets (Sottana et al., 2023; Goyal et al., 2023), rating them highly across all aspects of evaluation. This preference reveals the poor quality of reference summaries, raising concerns about the effectiveness of fine-tuning models on reference summaries and their perturbed version.

2. Synthetic negative samples generated from common approaches often fail to replicate the actual errors observed in model-generated summaries. Furthermore, the error distributions in generated summaries can vary significantly across different domains, rendering synthetic negative sample generation approaches insufficient to cover the wide variety of error types (Goyal and Durrett, 2021).
3. Contrastive learning approaches (Cao and Wang, 2021; Tang et al., 2022; Zhang et al., 2023) lack utilization of detailed, span-level information that could potentially improve summary faithfulness more effectively (Goyal and Durrett, 2021). In cases of unfaithfulness within LLM-generated summaries, typically only a few specific text spans are unfaithful. Only these problematic spans should be specifically targeted as negative examples during model training.

To address the above problems, we propose to annotate span-level hallucinations of LLM-generated summaries and then update the model using this fine-grained, span-level information. Our contributions in this paper are threefold: (1) we construct a dataset that contains both faithful and unfaithful summaries labeled at the span level; (2) we compare the effectiveness of three approaches (gradient ascent, unlikelihood training and task vector negation) that use span-level information to improve the faithfulness of the model, and (3) we demonstrate that unlikelihood learning is the most effective of the three approaches we compared.

2 Related Work

2.1 Improving Summarization Faithfulness

A number of prior approaches to improving faithfulness focus on post-processing. Dong et al. (2020) uses QA span fact correction models to revise entities in generated summaries to boost factual consistency. Similarly, Cao et al. (2020) corrects factual errors in generated summaries by training a corrector model on artificially created error data.

Other prior works leveraged synthetic negative sample summaries to improve faithfulness. Cao and Wang (2021) surveyed common errors that summarization models tend to make and designed strategies for constructing negative samples (e.g., entity swap, mask and regenerate) that corrupt the reference summaries. They then used contrastive learning to better discriminate between positive and negative examples, improving the representation and faithfulness during generation. Similarly, Tang et al. (2022) designed a linguistically informed taxonomy of factual errors for dialogue summaries and created synthetic negative samples based on the taxonomy before applying contrastive learning to improve faithfulness. Laban et al. (2023) proposed a cost effective protocol to create more natural sounding and manually verified synthetic negative samples, which can be used as training data for contrastive learning.

Chen et al. (2021) proposes to generate multiple contrastive candidate summaries featuring different entities from the source document, subsequently employing a discriminative model trained to differentiate between faithful summaries and synthetic negative ones, effectively ranking the candidates. Goyal and Durrett (2021) tried to tackle the problem from the perspective of training data. They demonstrated an approach to improve faithfulness by identifying unsupported facts in the training summaries and ignore the corresponding tokens

during training. Goyal et al. (2022) took a closer look at the training dynamics and found that longer training on noisy datasets contributes to factual inconsistency. They show that using token subsampling to dynamically modify the loss computation during training, down weighting high-loss tokens, can substantially improve factual consistency.

Some more recent methods have focused on the critique-and-refine approach. Akyurek et al. (2023) implemented a reinforcement learning framework where a critic model provides feedback on generated summaries. The summarizer then refines its output based on this feedback, with the summarization task metric serving as a reward for the critic model. This process enables the critic to guide the summarizer towards improving specific performance metrics, fostering a cycle of continuous improvement.

2.2 Span-level Hallucination Labeling

A great amount of studies have been conducted on evaluation metrics for summarization faithfulness (Zhang* et al., 2020; Yuan et al., 2021; Liu et al., 2023; Zha et al., 2023). However, the study of span-level hallucination labeling remains relatively underexplored. Zhou et al. (2021) proposed a task to predict token-level hallucinations in summarization and machine translation and introduced methods to create and fine-tune on synthetic data to solve the task. Goyal and Durrett (2021) presented an approach to label hallucinations in summaries at the dependency arc level, providing more fine-grained information. Though not specific to summarization, Liu et al. (2022) introduced a token-level reference-free hallucination detection benchmark and created multiple baselines.

3 Methods

3.1 Problem Formulation

As illustrated in Figure 3, training data comprises of both positive and negative samples of document-summary pairs. For each positive sample, the document D has a corresponding positive summary S_p where S_p = $(x_{p1}, x_{p2}, ..., x_{pT})$. Similarly, each negative sample includes a document D paired with a negative summary S_n where S_n = $(x_{n1}, x_{n2}, ..., x_{nT})$. Here, x denotes a token in the summary, and T denotes the number of tokens in the summary. Note that not all x_n tokens are considered hallucinated. Therefore, we use H_n to denote the set of hallucinated tokens in a negative sample summary S_n and $H_n \subseteq S_n$.

3.2 Methods Studied

We study three methods that can take advantage of both positive faithful summaries and negative unfaithful summaries with span-level hallucination labels. To manage the influence of positive and negative samples, we introduce a hyperparameter $\epsilon \in [0, 1]$ to distribute the weights assigned to each type of summary. Specifically, ϵ specifies the weight of negative samples, and its complement $1 - \epsilon$ specifies the weight of positive samples.

Gradient Ascent Recent research by Yao et al. (2024) has demonstrated that unlearning undesirable behaviors in LLMs through gradient ascent is an effective alignment technique. This method has been used to steer LLM outputs away from harmful content, copyright infringements, and to reduce hallucinations. In our study, we apply gradient ascent to decrease hallucinated content in summarization by minimizing the probability of generating unfaithful tokens. This is achieved by reversing the sign of the cross-entropy loss. We define the gradient ascent loss as follows:

$$L_{ga} = \begin{cases} -(1-\epsilon) \sum\limits_{x_p \in S_p} \log p_\theta(x_p|.) & \text{if } S_p \\ \epsilon \sum\limits_{x_n \in H_n} \log p_\theta(x_n|.) & \text{if } S_n \end{cases}$$

Here, p_θ represents the model parameterized by θ. The loss computation varies depending on the sample type: for a positive sample S_p, the negative log-likelihood (NLL) loss is calculated with a weight of $(1 - \epsilon)$; for a negative sample S_n, the loss is computed as the negative NLL, multiplied by the weight ϵ.

Unlikelihood Training Unlikelihood training, initially introduced by Welleck et al. (2020), was designed to mitigate common issues such as dull and repetitive outputs from language models while maintaining perplexity. Subsequent studies (Li et al., 2020) have demonstrated its effectiveness in addressing problems like excessive copying from context, frequent word overuse, and logical inconsistencies in generated texts. The adaptability of this method allows for its application across various tasks, particularly in reducing undesirable outputs. In this method, the model is trained to assign lower probabilities to unwanted generations by maximizing the complement of the probability of generating

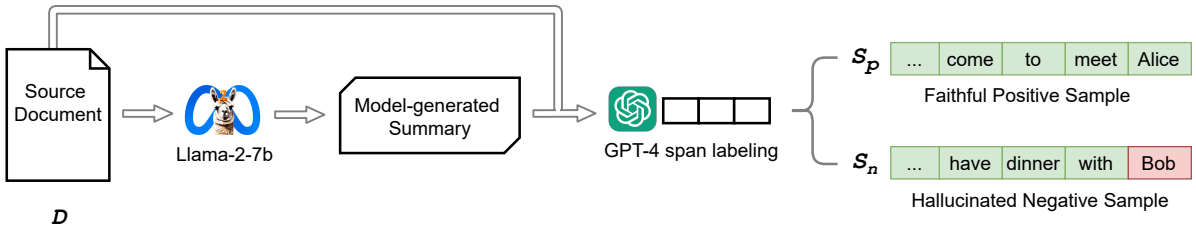

Figure 2: **Training data construction:** summaries of the source documents used for model training are generated using an LLM model (in our case Llama 2). Spans of text in the generated summaries that are unfaithful to the source document are automatically labeled using GPT-4 (using the prompt in Appendix A.3). Summaries that have no unfaithful spans labeled in their output are treated as positive training samples, and summaries that contain unfaithful spans are treated as negative training samples.

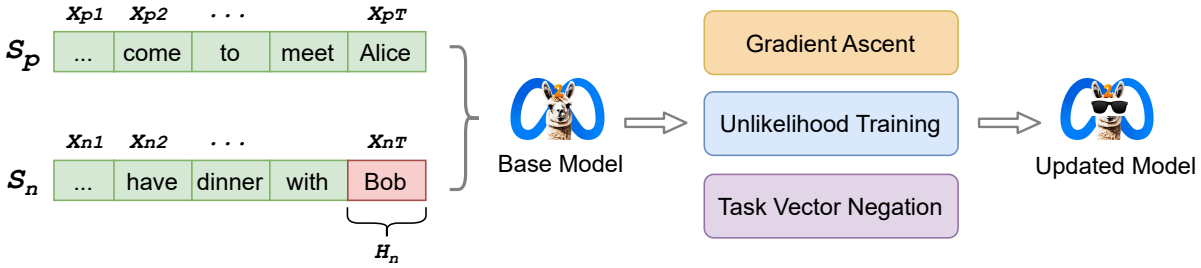

Figure 3: **Model update:** a base model is updated using both the faithful positive example summaries and the unfaithful negative example summaries with hallucination spans using one of three approaches we compare in this paper (1) Gradient Ascent, (2) Unlikelihood Training or (3) Task Vector Negation.

such tokens. For our specific application in summarization, we focus on minimizing the generation of hallucinated tokens. We define the unlikelihood loss as follows:

$$L_{ul} = \begin{cases} -(1-\epsilon) \sum\limits_{x_p \in S_p} \log p_\theta(x_p|.) & \text{if } S_p \\ -\epsilon \sum\limits_{x_n \in H_n} \log(1 - p_\theta(x_n|.)) & \text{if } S_n \end{cases}$$

This configuration computes the negative log-likelihood (NLL) for positive samples, weighted by $(1-\epsilon)$, and for negative samples, it calculates the NLL of the complement probability $(1 - p_\theta(x_n|.))$ weighted by ϵ. This approach ensures a strategic penalization of unfaithful tokens, thus aiming to enhance the overall faithfulness of the generated summaries.

Task Vector Negation Task vector arithmetic offers a straightforward method for modifying a pre-trained model to promote desired behaviors, as outlined by Ilharco et al. (2023). This technique involves calculating a task vector, τ_t, for a specific task t by subtracting the weights of the base pre-trained model (θ_{pre}) from the weights of a model that has been fine-tuned on that task (θ_{ft}). Formally, the task vector is defined as $\tau_t = \theta_{ft} - \theta_{pre}$. Building on this concept, Ilharco et al. (2023) demonstrated that negating a task vector fine-tuned on toxic language can effectively diminish the generation of toxic outputs. Inspired by this, we apply a similar approach to address the generation of unfaithful summaries. We define the resulting model's weights as:

$$\theta_{res} = \theta_{pre} + (1-\epsilon)\tau_{pos} - \epsilon\tau_{neg}$$

Here, τ_{pos} represents the task vector derived from fine-tuning on positive, faithful sample summaries, while τ_{neg} is obtained from fine-tuning on negative hallucinated summaries. This method allows us to manipulate the model composition to reduce the production of unfaithful summaries by strategically negating the influence of the undesired traits encoded in τ_{neg}.

4 Data

Recent studies (Sottana et al., 2023; Goyal et al., 2023) have found that even open-source LLMs outperform the gold standard reference summaries of large and well-established datasets. Human evaluators scored highly of LLM summaries on *relevance*, *fluency*, *coherence*, and *consistency*, all four common aspects of summarization evaluation, while rating the reference summaries as the worst on all

Source:

Pam: Hey Robert, you said you could help with Tom's birthday?
Robert: Sure, what do you need?
Pam: I have to go shopping, cook, and clean, and I figured out I don't have time to pick up the balloons.
Robert: From where?
Pam: There's this store in the city centre that sells these awesome floating balloons.
Robert: No problem, just text me the address.
Pam: Bless you!
Robert: ;)

Baseline Summary:
Pam asked Robert for help with Tom's birthday celebration, as she needs to go shopping, cook, and clean, and doesn't have time to pick up floating balloons from a store in the city centre. Robert agreed to help by providing the address of the store .

Gradient Ascent Summary:
Pam asked Robert for help with Tom's birthday celebration, including picking up floating balloons from a store in the city centre. Robert agreed to help and requested the store's address.

Unlikelihood Summary:
Pam asked Robert if he could help with Tom's birthday celebration, specifically asking for his assistance in picking up floating balloons from a store in the city centre. Robert agreed to help and requested the store's address.

Task Vector Negation Summary:
Pam asked Robert for help with Tom's birthday celebration, including shopping, cooking, and cleaning. Robert agreed to help and Pam provided the address of a store in the city centre where she needed him to pick up floating balloons .

Figure 4: An example of SAMSum dialogue and its corresponding summaries generated from models fine-tuned with four approaches. Unfaithful spans are highlighted in pink .

metrics, indicating the poor quality of reference summaries in publicly available datasets. Additionally, previous approaches of synthetically constructing negative samples with strategies inspired from common types of errors (Cao and Wang, 2021; Tang et al., 2022; Zhang et al., 2023), do not align with the actual errors made by generation models (Goyal and Durrett, 2021). Moreover, the actual error distributions vary significantly across different summarization domains. Thus, generating synthetic negative samples from reference summaries is both poor in quality and insufficient in coverage. To combat these two limitations, we construct a new dataset consisting of LLM-generated summaries that contain hallucinations labeled at the span level.

		Pos	Neg	Avg. hallu toks
Train	SAMSum	2774	2774	6.5%
	CNN	2774	2774	2.7%
Test	SAMSum	50	50	6.7%
	CNN	50	50	2.5%

Table 1: Statistics of constructed dataset. Each category of training set consists of 2774 samples, and the corresponding test set consists of 50 samples. On average, 6.5% of tokens in SAMSum summaries are unfaithful, and that for CNN summaries is 2.7%.

4.1 Dataset Construction

The dataset construction process is illustrated in Figure 2. We construct the training and test set from CNN (Nallapati et al., 2016) and SAMSum (Gliwa et al., 2019), each belonging to news and dialogue, two of the most studied domains of summarization research. Summaries of their source documents were generated using `Llama-2-7b-chat` (Touvron et al., 2023) with top_p = 0.7 and temperature = 0.01 to minimize randomness. The prompts used to generate summaries are in A.1 and A.2).

Sottana et al. (2023) shows that GPT-4 (OpenAI et al., 2024) correlates highly with human judgments when acting as a reviewer. Expanding upon this prior work we use GPT-4 to label LLM-generated summaries to identify "spans of text that are inconsistent with the source document." The prompt we use to label these spans is provided in A.3). When a summary has no labelled span in it's output, it is considered a positive example of a summary; if a span has been labeled as "inconsistent with the source document" the summary is treated as a negative example.

After filtering out noisy outputs and balancing the data between positive and negative examples, we obtained 11,096 training samples and 200 test samples, half of which come from the CNN dataset and the other half from SAMSum. Positive and negative examples were selected to provide an even 50/50 distribution across the data as shown in Table 1. On average, SAMSum has a higher rate of hallucinated tokens at 6.5% while CNN has a hallucinated token rate of 2.7%.

5 Experimental Setup

5.1 Model and Training

All experiments are performed using the Huggingface implementation of **Llama2-7b**

(`meta-llama/Llama2-7b-chat-hf`)[1]. For efficient fine-tuning, we adopt Low-Rank Adaptation (LoRA Hu et al., 2022) and apply low-rank update matrices to all linear modules, with rank $r = 64$, $\alpha = 128$, and dropout probability of LoRA layer = 0.05. With these settings, 2.3% of the model parameters are trainable. The training processes use an AdamW optimizer with a learning rate set to 5e-5. We train all models on two RTX 6000 Ada GPUs. To avoid out-of-memory errors, we set step size = 1, use bf16 precision, gradient checkpointing, and gradient accumulation steps = 8, making the effective batch size 16, and we train for 1 epoch on the training set for each experiment.

The baseline for comparison is a Llama2-7b model fine-tuned only on the positive portion of training data with regular cross-entropy loss, with no information from negative samples.

Figure 4 shows an example SAMSum dialogue and its corresponding summaries generated from the baseline model and three other models each fine-tuned using one of the methods studied in this paper.

5.2 Automatic Evaluation

For evaluation, we use the updates models to generate summaries (with the same settings as in Section 4.1) on the test set and employ the same labeling process to label hallucination spans with GPT-4 as in Section 4.1. Summaries that were incomplete or degenerate into repetitions (Holtzman et al., 2020) were automatically detected and considered as unfaithful summaries. We note this method as **GPT4SL** (GPT-4 span labelling) in our results.

In addition to GPT4SL, we employ two reference-free metrics to assess the quality of the generated summaries G-Eval and AlignScore. **G-Eval** (Liu et al., 2023) is a GPT-based metric that leverages Chain-of-Thought (CoT Wei et al., 2022) prompting to generate evaluation steps before computing an evaluation score. We use GPT-4 as the LLM for this metric and report the its consistency - the factual alignment between the summary and the source (Fabbri et al., 2021) in our results. **AlignScore** (Zha et al., 2023) is a non-LLM-based factual consistency metric based on a unified text-to-text information alignment function. This metric measures the factual alignment between the summary and the source text. Both metrics have demonstrated high correlations with human judgments, providing reliable automated evaluation.

[1]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

	GPT4SL	G-Eval	AlignScore
SAMSum	67.0	4.631	0.696
CNN	69.0	4.897	0.800

Table 2: Average scores of the baseline model according to GPT-4 span labeling (GPT4SL), G-Eval, and A-Score. GPT-4 span labeling is computed as the percentage of summaries that are faithful (no hallucinations).

5.3 Human Verification

To further validate the results, we performed human verification on the summaries generated by the baseline and unlikelihood model for the SAMSum testset. We provided annotators with a simple guideline that stated:

> *if a span contains inconsistent/contradictory information to the source, then label as hallucination; if a span contains new information not present in the source and cannot be verified by a Google search, then label as hallucination.*

This way, we do not falsely label true information that provides additional world knowledge that is important for summary comprehension (Cao et al., 2022). Labeling was performed in two passes, an initial pass was performed by a single annotator for each summary, and then a second validation pass was performed where the team of annotators reviewed the labeling of each summary together.

6 Results and Discussion

6.1 Automatic Evaluation Results

The results for the baseline system, in which the model has only been updated using faithful summaries (S_p) are shown in Table 2. The results of the three model update methods explored in this paper are shown in Table 3 for different values of ϵ. Results are presented for three automatic metrics, GPT-4 span labeling (GPT4SL), G-Eval and AlignScore (A-Score). For the SAMSum dataset, the unlikelihood model with $\epsilon = 0.1$ improves the GPT4SL metric from 67% to 80%, which correlates to reducing the number of summaries with hallucinations from 33 down to 20, a 39% reduction. On the CNN dataset, the unlikelihood model increases the GPT4SL score from 69% to

	ϵ	Gradient Ascent			Unlikelihood			Task Vector		
		GPT4SL	G-Eval	A-Score	GPT4SL	G-Eval	A-Score	GPT4SL	G-Eval	A-Score
SAMSum	0.1	64.0	4.63	0.6867	**80.0**†	4.63	**0.7403**†	65.0	4.56	0.7127
	0.3	13.0†	2.96†	0.6964	71.0	4.70	0.7331	68.0	4.51	0.7165
	0.5	0.0†	2.26†	0.4862†	76.0	4.71	0.7380	69.0	4.54	0.7293
	0.7	0.0†	2.49†	0.4885†	67.0	**4.72**	0.7394†	68.0	4.60	0.7184
	0.9	0.0†	1.13†	0.3710†	57.0	4.10†	0.7355	69.0	4.53	0.7083
	1.0	-	-	-	-	-	-	72.0	4.58	0.7208
CNN	0.1	52.0†	4.59†	0.7662	**87.0**†	4.90	0.8151	73.0	4.84†	0.7909
	0.3	2.0†	2.69†	0.7982	83.0†	**4.91**	0.8129	64.0	4.86	0.7891
	0.5	0.0†	2.25†	0.7114†	72.0	4.84	**0.8295**	73.0	**4.91**	0.7852
	0.7	0.0†	2.44†	0.721†	57.0	4.59†	0.7942	**87.0**†	4.86	0.7586†
	0.9	0.0†	0.69†	0.1661†	45.0†	3.81†	0.7894	84.0†	4.89	0.7950
	1.0	-	-	-	-	-	-	83.0†	4.87	0.7940

Table 3: Average summary-level scores of three studied methods according to GPT-4 span labeling (GPT4SL), G-Eval, and AlignScore (A-Score) with different ϵ values. GPT-4 span labeling is computed as the percentage of faithful summaries (no hallucinations). The best result in each metric is **in-bold**. Results that are statistically significantly different from the baseline ($p < 0.05$) are indicated with †.

	SAMSum	CNN
Baseline	4.22	1.29
Gradient Ascent	11.84	23.45
Unlikelihood	1.74	0.42
Task Vector	3.82	0.51

Table 4: Average percentage of tokens that are labeled as hallucinations by GPT-4 span labeling. Each value is the lowest number for that method across all ϵ values.

87%, which correlates to reducing the number of summaries with hallucinations from 31 down to 13, a 58% reduction. The best task vector negation model reduces SAMSum hallucinated summaries by 15%, lower than that of the unlikelihood approach, and it reduces CNN hallucinated summaries by 58%, matching the performance of the unlikelihood method. The gradient ascent models performed more poorly than the baseline due to the instability of training with gradient ascent. The outputs generated by the gradient ascent-trained models tended to degenerate into repetitions (especially for high ϵ), negatively impacting faithfulness.

The G-Eval and AlignScore metrics show a similar ranking of model performance, showing that the unlikelihood approach is the top performing method for the SAMSum dataset, and on the CNN dataset the task vector negation and likelihood models have similar performance.

Table 4 shows the average percentage of hallucinated tokens according to GPT-4 span labeling. The token-level results align with the summary-level results in Table 3, namely, unlikelihood is the most effective method, reducing hallucinated tokens by 59% on SAMSum and 67% on CNN, while task vector negation is less effective, reducing the number of hallucinated tokens by 9% on SAMSum and 60% on CNN. Gradient ascent negatively impacts performance, generating hallucinated tokens more than the baseline.

6.2 Human Verification Results

Following the protocol in Section 5.3, we manually annotated 200 SAMSum summaries: 100 from the baseline model and 100 from the unlikelihood model with $\epsilon = 0.1$. After excluding model refusals, 95 dialogues remained with summaries from both models.

Table 5 reports faithfulness results on these 95 paired examples. For G-Eval and AlignScore, we convert scalar scores into binary labels by selecting thresholds that maximize correlation with human judgments. Human annotations show that unlikelihood training increases faithful summaries from 57 to 67 and reduces hallucinated summaries from 38 to 28, a 26% reduction. This supports our automatic-metric findings that unlikelihood training reduces hallucinations.

All three automatic metrics also indicate fewer

	Baseline	Unlikelihood
Human	57	67
GPT4SL	64	75
G-Eval	77	80
AlignScore	50	67
	MCC with Human	
GPT4SL	0.669	0.572
G-Eval	0.373	0.543
AlignScore	0.043	0.088

Table 5: Results on 95 dialogues summarized by the baseline model and unlikelihood model with $\epsilon = 0.1$. The upper section is the number of summaries that are faithful according to each metric. G-Eval threshold = 4.7 and AlignScore threshold = 0.7. The lower section is Matthew's correlation coefficient (MCC) between automatic metrics and human judgments.

hallucinated summaries after unlikelihood training, with estimated reductions of 35% from GPT-4 span labeling, 16% from G-Eval, and 37% from AlignScore. Compared with human annotations, however, these estimates differ by roughly 10%.

To assess metric reliability, we compute Matthews correlation coefficient (MCC) against human labels, shown in the lower part of Table 5. GPT4SL achieves the strongest correlation, with MCC scores of 0.669 on baseline summaries and 0.572 on unlikelihood-trained summaries, outperforming G-Eval and AlignScore. This supports the effectiveness of GPT4SL for evaluating summary faithfulness.

At the token level, unlikelihood training reduces hallucinated tokens from 5.09% to 3.95%. GPT4SL obtains a token-level MCC of 0.553 against human annotations.

6.3 Discussion

ϵ and Model Performance: One prominent phenomenon we observe in Table 3 is the negative correlation between ϵ and model performance. In gradient ascent and unlikelihood training, the general trend is the higher the weight to put on negative samples the worse the model would perform. We manually verified some sample generations and found high ϵ values decimate a models language modeling ability, rendering its generation useless.

GPT-4 as a span labeler: The improved summary faithfulness shown in Section 6.1 combined with the high correlations with human judgements shown in Section 6.2 shows the effectiveness of using GPT-4 as a span labeler to automatically identify hallucinations in LLM-generated summaries. The token-level Cohen's κ agreement between human labels and GPT4SL is 0.544, which indicates moderate agreement. In future work, we plan to investigate the effectiveness of GPT-4 to automatically label spans of hallucinated text in LLM-generated summaries and explore it's potential as a fine-grained faithfulness metric.

The instability of gradient ascent: In this work we observed that gradient ascent is prone to instability during training, indicated by its jagged loss curves (shown in Appendix B) and it's sensitivity to the hyperparameter ϵ. In our experimentation, the majority of the models trained with gradient ascent could not generate complete summaries and thus this approach performed more poorly than the baseline on both the SAMSum and CNN datasets. Further investigation into the cause of gradient ascent's instability is required.

Comparison with Contrastive Learning: Cao and Wang (2021) uses summary-level faithfulness information to improve faithfulness and find contrastive learning works the best. Being more fine-grained, our work focuses on leveraging span-level information. However, to perform contrastive learning on span (token)-level, we would need a corresponding positive token for each negative token, which our current dataset does not have. As a future line of work, we will investigate means of making contrastive learning work on the token-level.

7 Conclusion

In this study, we introduce a novel approach to improve summarization faithfulness by fine-tuning an LLM on a dataset of automatically labeled negative sample summaries. Our methodology involves constructing a dataset with both positive and negative samples, achieved by first generating summaries using an LLM, followed by labeling span-level hallucinations in these summaries with GPT-4. We then study three fine-tuning methods: *gradient ascent*, *unlikelihood training*, and *task vector negation* that can take advantage of both positive and span-level negative samples. Our findings suggest that unlikelihood training is particularly effectively at using span-level annotations to reduce the occurrence of unfaithful summaries, thereby enhancing summary faithfulness.

8 Limitations

This work explored the effectiveness of span-labeling in combination with *gradient ascent*, *unlikelihood training*, and *task vector negation* to reduce the occurrence of hallucinations in LLM-generated summaries. The majority of the evaluations in this work were performed with automatic metrics, including metrics proposed in prior works (G-Eval (Liu et al., 2023) and AlignScore (Zha et al., 2023)) and a novel metric used in this paper GPT4SL (GPT-4 span labelling) which was required to identify specific spans of text in the generated summary that were likely unfaithful to the source document.

To evaluate the performance of these metrics we manually created a small set of gold annotations for summaries generated with the baseline and unlikelihood approaches for the SAMSum testset. Due to limited resources we did not create similar annotations for task vector negation or the CNN testset. Although we have shown that the automatic metrics used in this work correlate well with the human labels, we observe that both the GPT4SL and G-Eval metrics tend to underestimate the true occurrence of hallucinations. Additionally, during the annotation of the SAMSum dataset we found that a number of summaries were very difficult to label as the situational context in which the dialog occurred could mean that a phrase in the summary could be interpreted as being either faithful or unfaithful to the source.

Although the metrics and manual annotation used in this work are not the main contribution of this paper, we acknowledge that a more rigorous evaluation of these metrics are required to support our findings. We intend to do this in future work.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.

Meng Cao, Yue Dong, and Jackie Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2023. News summarization and evaluation in the era of gpt-3.

Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. Training dynamics for text summarization models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2061–2073, Dublin, Ireland. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

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

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

Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on large language model hallucination via a creativity perspective.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676, Singapore. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

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

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: Reliably evaluating large language models on sequence to sequence tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8776–8788, Singapore. Association for Computational Linguistics.

Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5657–5668, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu,

Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng 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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Nan Zhang, Yusen Zhang, Wu Guo, Prasenjit Mitra, and Rui Zhang. 2023. FaMeSumm: Investigating and improving faithfulness of medical summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10915–10931, Singapore. Association for Computational Linguistics.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

A LLM prompts used in experimentation

A.1 Prompt for LLM-generated summarization - SAMSum dataset

System prompt: You are an accurate summarizer that always writes concise summaries of text that is as consistent with the source text as possible. You will be given a dialogue conversation in the user prompt, and you need to provide a concise summary of the conversation in 2 sentences. Do not start your response with "Sure".

User prompt: {source_doc}

A.2 Prompt for LLM-generated summarization - CNN dataset

System prompt: You are an accurate summarizer that always writes concise summaries of text that is as consistent with the source text as possible. You will be given a piece of news article in the user prompt, and you need to provide a concise summary of the article as a response. Do not start your response with "Sure".

User prompt: {source_doc}

A.3 Prompt for GPT-4 Span Labeling (GPT4SL)

System prompt: You will be given a source text and a summary of that text. Your job is to identify spans of text in the summary that is inconsistent to the source text.

The source text and summary are wrapped in XML tags like so: <source>source text</source> and <summary>summary text</summary>, and you should return the summary with the inconsistent part wrapped in XML tags like this: <summary>This is a summary with <hallu>inconsistent or hallucinated text</hallu></summary>.

If there are no inconsistencies in the summary, simply return the summary.

User prompt: <source>{source_doc}</source>
<summary>{model_summ}</summary>

B Loss Curves from Model Training

Figure 5: Unlikelihood $\epsilon = 0.1$ loss curve

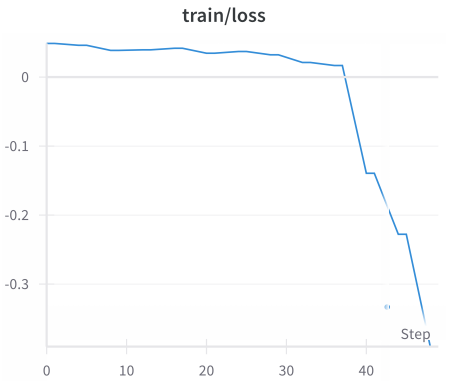

Figure 6: Gradient ascent $\epsilon = 0.1$ loss curve

Figure 7: Unlikelihood $\epsilon = 0.3$ loss curve

Figure 8: Gradient ascent $\epsilon = 0.3$ loss curve

Figure 9: Unlikelihood $\epsilon = 0.5$ loss curve

Figure 10: Gradient ascent $\epsilon = 0.5$ loss curve