Pre-trained Language Models Return Distinguishable Probability Distributions to Unfaithfully Hallucinated Texts

Taehun Cha and **Donghun Lee***
Department of Mathematics
Korea University
{cth127, holy}@korea.ac.kr

Abstract

In this work, we show the pre-trained language models return distinguishable generation probability and uncertainty distribution to unfaithfully hallucinated texts, regardless of their size and structure. By examining 24 models on 6 data sets, we find out that 88-98% of cases return statistically significantly distinguishable generation probability and uncertainty distributions. Using this general phenomenon, we showcase a hallucination-reducing training algorithm. Our algorithm outperforms other baselines by achieving higher faithfulness metrics while maintaining sound general text quality measures.[1]

1 Introduction

Hallucination is one of the key phenomena that undermine the reliability of large language models (LLMs), which recently gained large popularity in real-world applications (Zhang et al., 2023). Ji et al. (2023) characterized hallucinations with two perspectives: faithfulness and factuality. The former represents consistency to the provided source text, while the latter is consistency to the world knowledge. For example, if a user asks a machine to recommend a dinner menu and a machine answers that 'Cereal is a breakfast menu enjoyed by many people', then the answer is factual but not faithful to the user's request.

Before the pre-trained language model (PLM) era, researchers found out that generation probability and uncertainty measured by a language model are correlated with the faithfulness of a text (Kang and Hashimoto, 2020 and Xiao and Wang, 2021). Though their work utilized un-pre-trained models trained on specific tasks, like image captioning, these works hinted at PLMs' potential to distinguish unfaithfulness.

In this paper, we examine three research hypotheses first. (1) Does the unfaithfulness distinguishing ability generalize to the various sizes and types of PLMs? (2) Does the model size affect the ability? (3) How does the fine-tuning affect the ability? We examine these hypotheses with 24 pre-trained language models of various sizes and types on 6 data sets. From massive experiments, 88-98% cases return significantly distinguishable generation probability and uncertainty distributions. Using this phenomenon, we showcase a simple training algorithm that effectively reduces hallucination.

2 Related Works

Generation Probability/Uncertainty for Unfaithfulness Reduction

While training, Kang and Hashimoto (2020) reported truncating high-loss data points returns more faithful news titles. The result implies training on a data point with a high loss (i.e. low generation probability) can make a model generate unfaithful texts. While decoding, Xiao and Wang (2021) showed that the model's predictive uncertainty shows a positive correlation with unfaithfulness in an image captioning task. Though their work did not cover the PLMs, it hinted at the relationship between generation probability/uncertainty and faithfulness. Wan et al. (2023) extended this line of work when fine-tuning PLMs. But their (un)certainty is computed from fine-tuned models, not a PLM itself, without verifying the fine-tuning effect.

LLM Probability/Uncertainty as a Factuality Measure

As LLMs showed impressive performance on various tasks, hallucination researchers eagerly adopted LLMs in their works. Manakul et al. (2023) and Azaria and Mitchell (2023) reported LLMs' generation probability correlates well with factuality. Varshney et al. (2023) utilized LLMs'

* corresponding author

[1]Source codes are available on https://github.com/AIML-K/HalluDist

Findings of the Association for Computational Linguistics: EACL 2024, pages 12630–12639

generation probability to detect factually wrong texts. However, their works concentrated on the factuality of generated texts, not faithfulness. Moreover, they only utilized GPT3-like LLMs without verifying the size effect. Our work focuses on faithfulness while verifying the size effect.

PLM as a Quality Measure

PLMs can be used to construct quantitative metrics for various NLP tasks. After Zhang et al. (2020) and Sellam et al. (2020) introduced the BERTScore and BLEURT to compare generated and target texts, Yuan et al. (2021) introduced the BARTScore to measure the generated text quality. Yuan et al. (2021) showed that BART's generation probability shows a positive correlation with various quality measures, like informativeness or coherence. Our work is an extension and generalization of this work especially for the unfaithful hallucination.

3 Suggested Metrics

Notations

Let $D = \{(x_i, y_i, h_i)\}_{i=1}^{N}$ be a data set. For i^{th} reference text x_i, let $y_i = (y_{i,1}, y_{i,2}, ...y_{i,n_i})$ be a corresponding target text, where $y_{i,j}$ represents j^{th} token of i^{th} target text. Define $y_{i,!j} = (y_{i,1}, y_{i,2}, ..., y_{i,j-1}, [\text{MASK}], y_{i,j+1}, ..., y_{i,n_i})$, where a j^{th} token is replaced with a [MASK] token, and $y_{i,<j} = (y_{i,1}, y_{i,2}, ..., y_{i,j-1})$, a truncated target text. $h_i \in \{\text{Hallucinated, Entailed}\}$ is a unfaithful hallucination label. If the content of y_i is *faithful* to the content of x_i, then $h_i =$ Entailed. On the other hand, if the content of y_i is *unfaithful* to the content of x_i, then $h_i =$ Hallucinated. For the convenience of notations, let $D_{\text{Hallucinated}}$ be a subset of the data set D such that $h_i =$ Hallucinated. Likewise, define D_{Entailed} similarly.

Let f be a PLM. f can be an encoder model pre-trained on a masked language modeling task like BERT, a decoder model pre-trained on an autoregressive language modeling task like GPT2, or an encoder-decoder model like T5. For an encoder model, $f(x_i, y_{i,!j})[v] \in [0, 1]$ is a probability of a token v at masked position j given reference text and masked target text. So $f(x_i, y_{i,!j})[y_{i,j}]$ is the probability for the right token. For decoder and encoder-decoder models, $f(x_i, y_{i,<j})[v] \in [0, 1]$ is a probability of a token v at truncated position j given reference text and truncated target text. Hence $f(x_i, y_{i,<j})[y_{i,j}]$ is the probability for the right token.

Metrics

Given a PLM f, we compute two metrics for each data point (x_i, y_i). These metrics, frequently used in the hallucination literature (Xiao and Wang, 2021, Manakul et al., 2023, Varshney et al., 2023 and Wan et al., 2023), are as follow:

- Log Token Probability (**LogProb**): A metric used to estimate the given model's generation probability of a target text. We compute the mean of log token probabilities of a target text by $\frac{1}{n_i}\sum_{j=1}^{n_i} \log f(x_i, y_{i,<j})[y_{i,j}]$ when f is either a decoder or an encoder-decoder model, and $\frac{1}{n_i}\sum_{j=1}^{n_i} \log f(x_i, y_{i,!j})[y_{i,j}]$ when f is an encoder model.

- **Entropy**: A metric frequently used to estimate the given model's prediction uncertainty of a target text. We compute the mean of entropy of each token for a target text by $\frac{1}{n_i}\sum_{j=1}^{n_i} \mathbb{E}_{v\sim f(x_i, y_{i,<j})}[-\log f(x_i, y_{i,<j})[v]]$ when f is either a decoder or an encoder-decoder model, and $\frac{1}{n_i}\sum_{j=1}^{n_i} \mathbb{E}_{v\sim f(x_i, y_{i,!j})}[-\log f(x_i, y_{i,!j})[v]]$ when f is an encoder model.

After computing the metrics for each data point, we obtain a metric distribution $\mathbb{P}$ for a data set. With the $\mathbb{P}$, we can obtain an empirical cumulative distribution function (cdf), F.

4 Distribution Distinguishability

Let $\mathbb{P}$ be a distribution of a metric based on a PLM f, and $\mathbb{D}$ be a statistic computing distinguishability between two distributions. In this section, our goal is to verify (1) whether $\mathbb{D}(\mathbb{P}(D_{\text{Hallucinated}})||\mathbb{P}(D_{\text{Entailed}}))$ is statistically significant, and how the (2) model size and (3) fine-tuning of PLMs affect the distinguishability.

4.1 Experimental Setup

We utilize two statistics to quantify the distinguishability between two distributions.

- Kolmogorov–Smirnov Statistic (**KS statistic**, Kolmogorov, 1933): Given two cdfs, F_1 and F_2, the KS statistic is computed as $K(F_1, F_2) = \sup_x |F_1(x) - F_2(x)|$. Intuitively, K represents the maximum discrepancy between two cdfs. The KS statistic does not require distributional assumption unlike the t-test used in Wan et al. (2023).

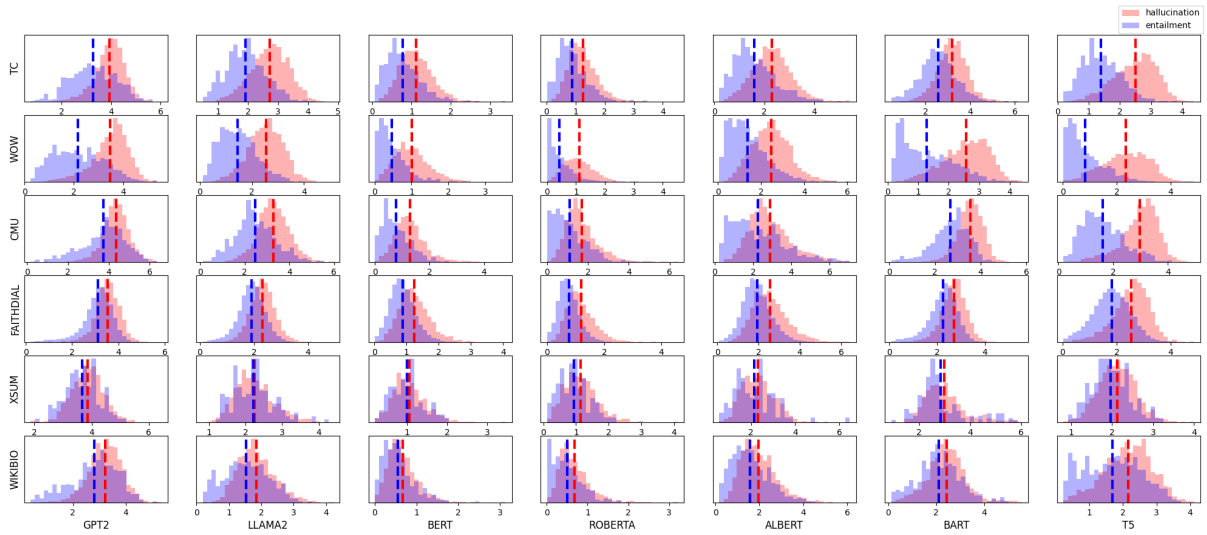

Figure 1: Empirical Entropy distribution and mean of $D_{Hallucinated}$ and $D_{Entailed}$ for each model and data set. We first compute Entropy for each data point, then separate the points according to the hallucination label. x-axis represents Entropy and y-axis represents the relative frequency. We plot the result of the smallest models for each model type.

- **Wasserstein Distance** (Kantorovich, 1960): Given two one-dimensional cdfs, F_1 and F_2, the Wasserstein-1 distance is computed as $W(F_1, F_2) = \int_0^1 |F_1^{-1}(q) - F_2^{-1}(q)|dq$, which represent the discrepant area between the two cdfs.

Appendix A intuitively visualizes two metrics, given the two cdfs.

The KS statistic enables a non-parametric statistical test called the Kolmogorov–Smirnov test (KS test) on distributional differences, unlike the Wasserstein distance. However, the KS statistic is sensitive to differences between the modes of two distributions while insensitive to their tails (Lipp and Vermeesch, 2023). We mainly utilize the KS statistic to test the significance of distinguishability and use the Wasserstein distance to compare the overall difference between models.

For a comprehensive analysis, we gather several natural language generation data sets containing hallucination label h_i from multiple tasks. For the knowledge-grounded dialogue task, we utilize BEGIN data set (Dziri et al., 2022b) and FaithDial data set (**FaithDial**, Dziri et al., 2022a). The BEGIN data set consists of three subsets based on existing data sets: TopicalChat (**TC**, Gopalakrishnan et al., 2019), Wizard of Wikipedia (**WOW**, Dinan et al., 2019) and CMU Document Grounded Conversations (**CMU**, Zhou et al., 2018). For the summarization task, we use XSum Hallucination Data Set (**XSum**, Maynez et al., 2020), where human annotates unfaithful hallucination labels on machine-generated summaries. For Wiki-like text generation, we utilize SelfCheckGPT data set (**WikiBio**, Manakul et al., 2023) based on WikiBio data set (Lebret et al., 2016), where human annotates unfaithful hallucination labels on GPT-3 generated biography for corresponding Wikipedia page. In summary, we utilize 6 data sets. Basic statistics of each data are reported on Appendix B.1.

Our analysis requires white-box models returning full probability distribution of tokens. As a result, we utilize three general types of pre-trained open-source transformer models. For the decoder model, we test 4 sizes of **GPT2** (Radford et al., 2019) and 3 sizes of **Llama2** (Touvron et al., 2023). For the encoder model, we test 4 **BERT** (Devlin et al., 2019), 4 **ALBERT** (Lan et al., 2020) and 2 **RoBERTa** (Liu et al., 2020). For the encode-decoder model, we test 5 **T5** (Raffel et al., 2020) and 2 **BART** (Lewis et al., 2020). In summary, we test 24 models.

4.2 Does PLM return Distinguishable Distributions to Unfaithful Texts?

For each combination of a data set and a model, we implement the KS test on the KS statistics with a p-value of 0.01. We compute the mean and standard deviation of the KS statistic. The results are on Table 1.

Regardless of model type and metrics, PLMs return significantly distinguishable distributions

	LogProb		Entropy	
	Sig.	KS	Sig.	KS
Encoder	88.33% (53 / 60)	0.3144 (0.1097)	90.00% (54 / 60)	0.3274 (0.1279)
Decoder	**92.86%** (39 / 42)	**0.3686** (0.1536)	88.10% (37 / 42)	0.3492 (0.1589)
Enc-Dec	88.10% (37 / 42)	0.2652 (0.1187)	**97.62%** (41 / 42)	**0.3927** (0.1698)

Table 1: Summary table for the KS test. Sig. is a ratio of the statistically significant KS test with a p-value of 0.01. KS is the mean and standard deviation of the KS statistics.

for $D_{Hallucinated}$ and $D_{Entailed}$ for 88-98% cases. For LogProb, decoder models return more significant results. On the other hand, for Entropy, encoder-decoder models return more significant results.

Empirical Entropy distributions for each model and data set are presented on Figure 1. After computing an Entropy for each data point, we plot two histograms (in blue and red) with respect to the hallucination label. Most cases return distinguishable mean and distributions. PLMs tend to assign higher Entropy on $D_{Hallucination}$, representing higher uncertainty. Meanwhile, PLMs tend to assign higher LogProb on $D_{Entailment}$ as shown in Appendix C. Roughly speaking, PLMs are internally less confident and less certain when they predict hallucinated texts.

4.3 Model Size Effect

Multiple researchers reported that GPT3-like LLMs can distinguish hallucinated texts (Manakul et al., 2023 and Azaria and Mitchell, 2023). It is natural to ask whether the distinguishing ability is enhanced as its size grows. For comparison, we visualize Wasserstein distance of each metric relative to the smallest model's statistics. To see the trend, we inspect models with more than three size variations, GPT2, Llama2, ALBERT, and T5. We compute the mean of Wasserstein distance for all data sets. The results are on Figure 2.

The results show that **bigger size does not guarantee better distinguishability**. Notably for T5, the bigger model returns much less distinguishable distributions between hallucination and entailment groups. This tendency is observed through all metrics. Researchers should not blindly adopt LLMs to distinguish hallucinated texts without verifying their size effect.

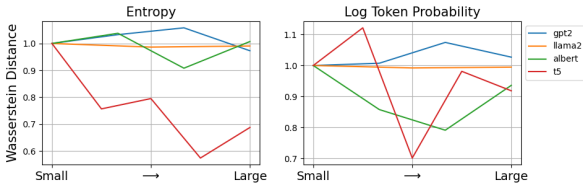

Figure 2: Visualization of the size effect. We divide all the Wasserstein distances with the distances from the smallest model to visualize the relative change as the size grows.

4.4 Fine-tuning Effect

Researchers utilized loss (Kang and Hashimoto, 2020) or uncertainty (Xiao and Wang, 2021) of the generation model once trained on target data to reduce unfaithfulness. It is also natural to ask how the fine-tuning of PLM on target data affects the distinguishability.

We train GPT2 models on the WOW and CMU training data set and check the statistics on the WOW and CMU portion of BEGIN data. Note that the training data does not contain the portion from the BEGIN data set so data contamination does not occur. The results for WOW data set are on Figure 3.

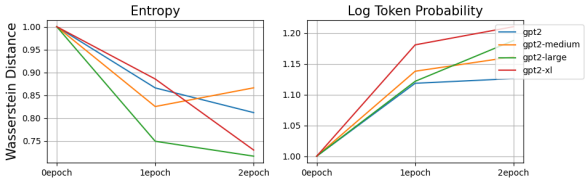

Figure 3: Fine-tuning effect for WOW data set. We divide all the Wasserstein distances with the distances from the pre-trained model to visualize the relative change as training proceeds.

The distinguishability from either metric is affected by fine-tuning while showing different trends. The distinguishability of LogProb increases as fine-tuning proceeds while the distinguishability of Entropy tends to decrease. We find similar trends in the CMU data set, as shown in Appendix D. Researchers should verify the fine-tuning effect of their target metric when they apply hallucination-reduction techniques.

5 Hallucination Reduction with Weighted Training

In this section, we showcase a weighted training method to mitigate hallucination. The idea is sim-

Data Set	Method	Q^2 F1	Q^2 NLI	SummaC	FactKB	ROUGE-L	BERT Score	BART Score
WOW	Unweighted	0.6521 (0.02)	0.6947 (0.02)	0.2941 (0.04)	0.5633 (0.03)	0.2862 (0.00)	0.3012 (0.00)	-2.7871 (0.01)
	CTRL	0.6746 (0.02)	0.7165 (0.01)	0.3051 (0.03)	0.5774 (0.01)	0.2741 (0.01)	0.3070 (0.01)	-2.7759 (0.02)
	Truncation	0.6996 (0.01)	0.7455 (0.01)	0.4089 (0.03)	0.6252 (0.02)	0.2788 (0.00)	0.3133 (0.00)	-2.7998 (0.02)
	mFACT	0.7539 (0.01)	0.7930 (0.01)	0.4988 (0.04)	0.6966 (0.03)	0.3068 (0.00)	**0.3367** (0.00)	-2.8348 (0.04)
	Ours-LogProb	0.7689 (0.02)	0.7946 (0.02)	0.4287 (0.04)	0.7033 (0.03)	0.2960 (0.01)	0.2963 (0.01)	**-2.7633** (0.05)
	Ours-Entropy	**0.7742** (0.02)	**0.8040** (0.01)	**0.5503** (0.02)	**0.7273** (0.01)	**0.3105** (0.00)	0.3124 (0.00)	-2.7811 (0.02)
FaithDial	Unweighted	0.7830 (0.03)	0.8439 (0.02)	0.1761 (0.05)	0.6156 (0.04)	0.3066 (0.00)	0.3360 (0.00)	-2.7874 (0.02)
	CTRL	0.7758 (0.01)	0.8405 (0.01)	0.2255 (0.05)	0.6267 (0.04)	0.2921 (0.00)	0.3384 (0.00)	-2.7769 (0.04)
	Truncation	0.7804 (0.01)	0.8479 (0.01)	0.3055 (0.06)	0.6205 (0.02)	0.2938 (0.00)	0.3369 (0.00)	-2.7903 (0.04)
	mFACT	0.8108 (0.0)	0.8733 (0.0)	**0.4099** (0.04)	0.6885 (0.02)	0.3023 (0.00)	**0.3460** (0.00)	-2.8402 (0.04)
	Ours-LogProb	**0.8454** (0.02)	0.8841 (0.02)	0.3652 (0.10)	**0.7706** (0.04)	0.3135 (0.01)	0.3371 (0.00)	-2.7251 (0.03)
	Ours-Entropy	0.8403 (0.02)	**0.8905** (0.01)	0.4092 (0.07)	0.7475 (0.03)	**0.3179** (0.00)	0.3401 (0.00)	**-2.7166** (0.02)
MediQA	Unweighted	0.7912 (0.01)	0.8333 (0.01)	0.5152 (0.02)	0.9987 (0.00)	0.2491 (0.01)	0.1712 (0.01)	-2.8650 (0.03)
	CTRL	0.7754 (0.02)	0.8189 (0.02)	0.4899 (0.02)	**0.9988** (0.00)	0.2355 (0.01)	0.1602 (0.01)	-2.9055 (0.04)
	Truncation	0.7784 (0.01)	0.8180 (0.01)	0.5349 (0.02)	**0.9988** (0.00)	0.2364 (0.01)	0.1710 (0.01)	**-2.8126** (0.05)
	mFACT	0.7936 (0.02)	0.8334 (0.02)	0.5087 (0.02)	**0.9988** (0.00)	**0.2540** (0.01)	**0.1784** (0.00)	-2.8837 (0.04)
	Ours-LogProb	**0.8129** (0.02)	**0.8579** (0.02)	**0.5416** (0.01)	0.9927 (0.01)	0.2447 (0.01)	0.1748 (0.01)	-2.8680 (0.06)
	Ours-Entropy	0.7853 (0.02)	0.8371 (0.02)	0.4966 (0.01)	0.9984 (0.00)	0.2465 (0.00)	0.1701 (0.01)	-2.8530 (0.06)

Table 2: Comparison table of faithfulness metrics (left) and text quality metrics (right). We mark the best score in bold and the second best with an underline.

ple. As we observe in Section 4, a data point with high Entropy tends to contain a hallucination. Similarly, a data point with low LogProb tends to contain a hallucination. Then what would happen if we use Entropy or LogProb as a loss weight for training?

We compare four baseline training methods on three data sets: the usual **Unweighted** training, a control-token method (**CTRL**, Rashkin et al. (2021)), and other loss weighting methods (**Truncation** (Kang and Hashimoto, 2020) truncate high loss points and **mFACT** (Qiu et al., 2023) weight the loss with faithfulness score). We compare four faithfulness metrics and three general text quality metrics. A more detailed explanation is in Appendix E. The results are on Table 2.

For knowledge-grounded dialogue data sets, our algorithms, with both Entropy and LogProb, improve faithfulness compared to Unweighted by a large margin, in all cases. The same phenomenon occurs when compared with other baselines. It is interesting since CTRL is designed to mitigate hallucination, especially in the knowledge-grounded dialogue task.

For MediQA, Ours-LogProb outperforms Unweighted on most metrics. Ours-LogProb outperforms Truncation and mFACT on Q^2 and SummaC, though both are designed to reduce hallucination in the summarization task. The result shows not only powerful hallucination reduction performance compared to other task-specific methods but also the general applicability of our methods through various tasks.

Our algorithm maintains general text quality measures, which we do not directly target. More interestingly, our method often outperforms other baselines. It may indicate its potential for enhancing not only faithfulness but also the overall fidelity and quality of generated text across diverse evaluation metrics.

6 Conclusion

Our work is the first comprehensive analysis of the PLMs' unfaithful hallucination-distinguishing ability. We compare PLMs' generation probability and uncertainty distributions of unfaithful and entailed texts. Regardless of the model type and size, PLMs return statistically distinguishable distributions to unfaithfully hallucinated and entailed texts for 88-98% cases. Unlike usual practice, the smaller models show comparable (and sometimes better) distinguishability to the largest models, while the distinguishability of Entropy declines while LogProb increases after fine-tuning. Utilizing this phenomenon, we showcase a hallucination-reducing training algorithm that outperforms other baselines with hallucination reduction while maintaining sound general text quality measures. We hope these findings lead to a deeper understanding of the hallucination phenomenon and more reliable hallucination mitigating techniques.

Limitation

Though we made comparisons *within* each model, comparison *between* models raises a subtle issue. For example, GPT2 and Llama2 are all decoder models but utilize different tokenizers. The vocab size of GPT2 is 50,257 while Llama2 is 32,000. As a result, the overall token probability is lower and entropy is higher for GPT2 since it should consider many more tokens for generation. It makes cross-model comparison difficult and requires some generalized form of (un)certainty, which is beyond the scope of this work.

A PLM's unfaithful hallucination-distinguishing ability does not imply the faithfulness of the text generated from the PLM. Likewise, confidence or certainty computed from the inner state of a model does not imply the certainty presented in the generated text. As a result, a model can generate unfaithful text in a confident tone but with high entropy. Researchers should not be confused between the tone and computed uncertainty especially when they work with black-box LLMs like GPT3.

As shown on Figure 1, distinguishability is relatively vague for the XSum data set. One possible reason can be the length of the reference text. The mean length of reference text is 384 words, which is much longer than other data sets (at most 286 words). Though we do not inspect the reference text length effect on entropy and log token probability, further analysis is required.

Acknowledgements

This work is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1G1A1102828).

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022a. FaithDial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. 2022b. Evaluating attribution in dialogue systems: The BEGIN benchmark. *Transactions of the Association for Computational Linguistics*, 10:1066–1083.

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 933–952, Singapore. Association for Computational Linguistics.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

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 Comput. Surv.*, 55(12).

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

L. V. Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422.

A. Kolmogorov. 1933. Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari*, 4:83–91.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Alex Lipp and Pieter Vermeesch. 2023. The wasserstein distance as a dissimilarity metric for comparing detrital age spectra and other geological distributions. *Geochronology*, 5(1):263–270.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

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.

Yifu Qiu, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2023. 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, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–718, Online. Association for Computational Linguistics.

Max Savery, Asma Ben Abacha, Soumya Gayen, and Dina Demner-Fushman. 2020. Question-driven summarization of answers to consumer health questions. *Scientific Data*, 7(1):322.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Yixin Wan, Fanyou Wu, Weijie Xu, and Srinivasan H. Sengamedu. 2023. Sequence-level certainty reduces hallucination in knowledge-grounded dialogue generation.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

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

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*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang,

Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

A Visualization of KS Statistic and Wasserstein Distance

We visualize two statistics, the KS statistic and Wasserstein distance for one-dimensional cdfs in Figure 4.

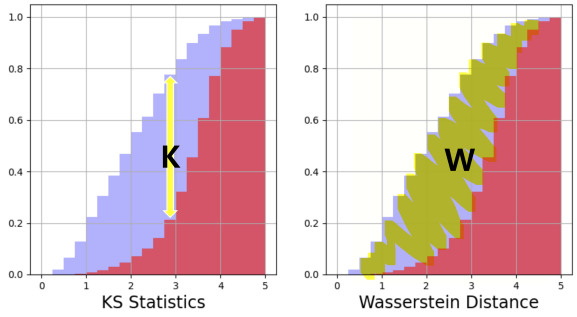

Figure 4: Visualization of the Kolmogorov–Smirnov statistic and Wasserstein distance. Red and blue histograms are separate cdfs to compare and the yellow arrow and the area represent each statistic.

B Basic Statistics of Data

B.1 Hallucination Data

On Table 3, we report the basic statistics of data utilized in Section 4.

B.2 Weighted Training Data

On Table 4 we report the basic statistics of data utilized on Section 5.

C Log Token Probability Distribution for Hallucinated and Entailed Data Sets

We present the log token probability distribution of $D_{Hallucinated}$ and $D_{Entailed}$ on Figure 6. Probability on $D_{Hallucinated}$ is relatively lower than $D_{Entailed}$, except BART. Roughly speaking, PLMs are internally more confident when they predict entailed texts.

D Fine-tuning Effect on CMU Data Set

We visualize the fine-tuning effect on the CMU data set on Figure 5. We can check a similar trend with the WOW data set shown in Section 4.4. Distinguishability with respect to entropy decreases while log token probability tends to increase.

Data Set	Train	Valid (Dev)	Test
TC	X	383 (305.11)	3,845 (301.49)
WOW	X	430 (55.09)	3,607 (55.19)
CMU	X	416 (225.68)	3,607 (226.14)
FaithDial	33,887 (110.18)	6,297 (112.04)	6,441 (110.41)
XSUM	X	996 (385.06)	996 (420.22)
WikiBio	X	954 (295.84)	954 (262.50)

Table 3: The number of data points and the average number of words of each data set.

Data Set	Train	Valid	Test
WOW	41,489 (97.15)	2,294 (96.78)	2,224 (95.55)
FaithDial	33,887 (110.18)	6,297 (112.04)	6,441 (110.41)
MediQA	578 (334.82)	29 (447.41)	45 (509.77)

Table 4: The number of data points and the average number of words of each data set.

E Experimental Detail for Weighted Training

Algorithm 1 depicts the algorithm in detail. In lines 4 and 6, we compute each metric as proposed in Section 4. In line 10, we apply the softmax function and multiply N to obtain the same scale of total loss as the unweighted loss.

Here are the detailed explanation on used baselines:

- **Unweighted**: Usual unweighted training. It is equivalent to Algorithm 1 with $W = \mathbf{1_N}$.
- **CTRL**: Rashkin et al. (2021) reported applying control tokens (<first-person>, <entailed>, <low-prec> etc.) improves faithfulness in

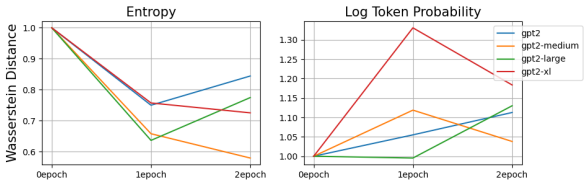

Figure 5: Fine-tuning effect for CMU data set. We divide all the Wasserstein distances with the statistics from the pre-trained model to visualize the relative change as training proceeds.

Algorithm 1 Weighted Training

Input: Training data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, Target model f, Pre-trained reference model g, Target metric $M \in \{\text{Entropy, LogProb}\}$, Weight vector $W = \phi$
for $i = 1$ **to** N **do**
  if $M = \text{Entropy}$ **then**
    $w_i = -M(g(x_i, y_i))$
  else if $M = \text{LogProb}$ **then**
    $w_i = M(g(x_i, y_i))$
  end if
  $W \leftarrow W \bigcup \{w_i\}$
end for
$W \leftarrow \text{SoftMax}(W) \times N$
Train f with $w_i \text{Loss}(x_i, y_i)$

knowledge-grounded dialogue. To obtain the control tokens, outer NER or NLI modules are required.[2]

- Loss Truncation (**Truncation**): Kang and Hashimoto (2020) suggested to truncate high-loss data points while training to achieve more faithful summarization. It is equivalent to Algorithm 1 if $w_i = 1$ for low-loss data points and $w_i = 0$ for the high-loss data points.[3]

- **mFACT**: Qiu et al. (2023) proposed another weighted-training method. They weigh the loss of each training example by its faithfulness score computed by a model trained on hallucination data sets. Though they proposed their method in a multi-lingual setting, it can be easily adapted to English.[4]

For the experiment, we use T5-small as g and f for our algorithm. For baselines, we train T5-small (about 60 million parameters) with an AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 1e-4 for all methods. For the other hyper-parameters, we only utilize the default setting of packages and repositories. We do not perform a hyper-parameter search. We train each model 5 times and report the mean and standard deviation of each metric. We use a machine with AMD Ryzen 9 5900X 12-Core Processor CPU with one NVIDIA RTX 3090 GPU.

Since our goal is to check the hallucination reduction performance of each training method, we only utilize training techniques from baselines, not decoding techniques (except attaching '<no-first-person> <entailed> <high-prec>' in CTRL decoding). While decoding, we use greedy deterministic decoding to exclude external factors.

CTRL is specialized in the knowledge-grounded dialogue task while Truncation and mFACT are specialized in the summarization task. So we use **WOW** (Dinan et al., 2019) and **FaithDial** (Dziri et al., 2022a) as benchmarks for the knowledge-grounded dialogue task and **MediQA-AnS**[5] (Savery et al., 2020) for the summarization task. Basic statistics of each data are reported on Appendix B.2.

MediQA-AnS data set is a question-driven summarization data set consisting of (question, reference, answer (summary)) tuples. The reference is a crawled web page and the answer is a human-written summary of the reference for the question. MediQA-AnS targets consumer-level questions on healthcare information. Since hallucinations in healthcare-related generations can severely harm human health, we select MediQA-AnS as a suited benchmark. We use MediQA-AnS as a training set and use the NAACL-BioNLP 2021 - Task 2 data set as the test set, which covers the same task.[6]

For faithfulness evaluation, we utilize three metrics. $\mathbf{Q^2}$ (Honovich et al., 2021)[7] first generates questions and answer candidates from the generated result. Then Q^2 applies a QA model on the reference to solve the generated question. After obtaining the answer from the reference, Q^2 compares it with the answer candidates lexically (**F1**) and semantically (**NLI**). Q^2 has been utilized as a de facto method to measure hallucination. Also, we use **SummaC** score (Laban et al., 2022) and **FactKB**'s probability of entailment (Feng et al.,

[2]We utilize the CTRL implementation from https://github.com/McGill-NLP/FaithDial
[3]https://github.com/ddkang/loss_dropper
[4]https://huggingface.co/yfqiu-nlp/mFACT-en_XX
[5]https://osf.io/fyg46/
[6]https://github.com/abachaa/MEDIQA2021/blob/main/Task2/README.md
[7]https://github.com/orhonovich/q-squared

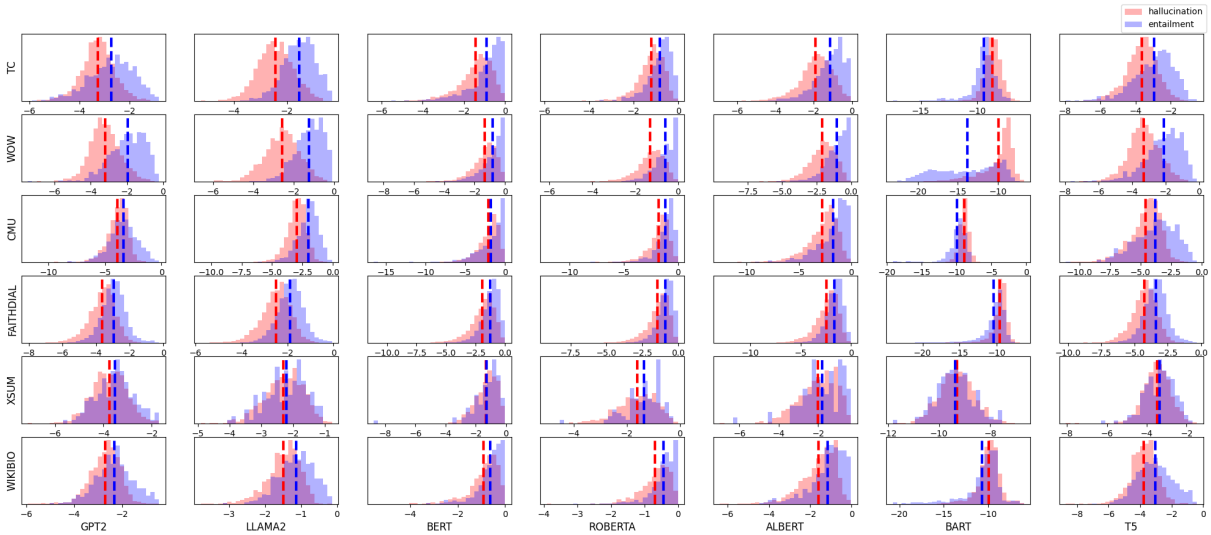

Figure 6: Empirical log token probability distribution and mean of $D_{Hallucinated}$ and $D_{Entailed}$ for each model and data set.

2023) as additional faithfulness metrics.

For general text quality evaluation, we utilize **ROUGE-L**[8], **BERTScore** (Zhang et al., 2020)[9], and **BARTScore** (Yuan et al., 2021)[10]. We train each model 5 times and compute the mean and standard deviation of each score on the test set.

[8]https://pypi.org/project/rouge/
[9]https://github.com/Tiiiger/bert_score
[10]https://github.com/stanfordnlp/string2string