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

Taehun Cha and Donghun Lee

Korea University

Department of Mathematics

Problem Definition

- Hallucination
  - Factuality: consistency to the world knowledge
  - Faithfulness: consistency to the provided source text

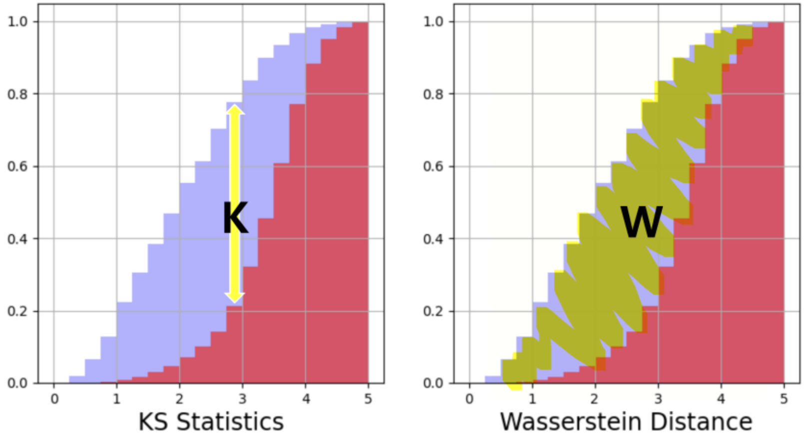
- Trained model generation probability and uncertainty showed correlation with the faithfulness of a text
  - Improved natural language generation via loss truncation (ACL 2020)
  - On hallucination and predictive uncertainty in conditional language generation (EACL 2021)

- How about PLM itself?

Problem Definition

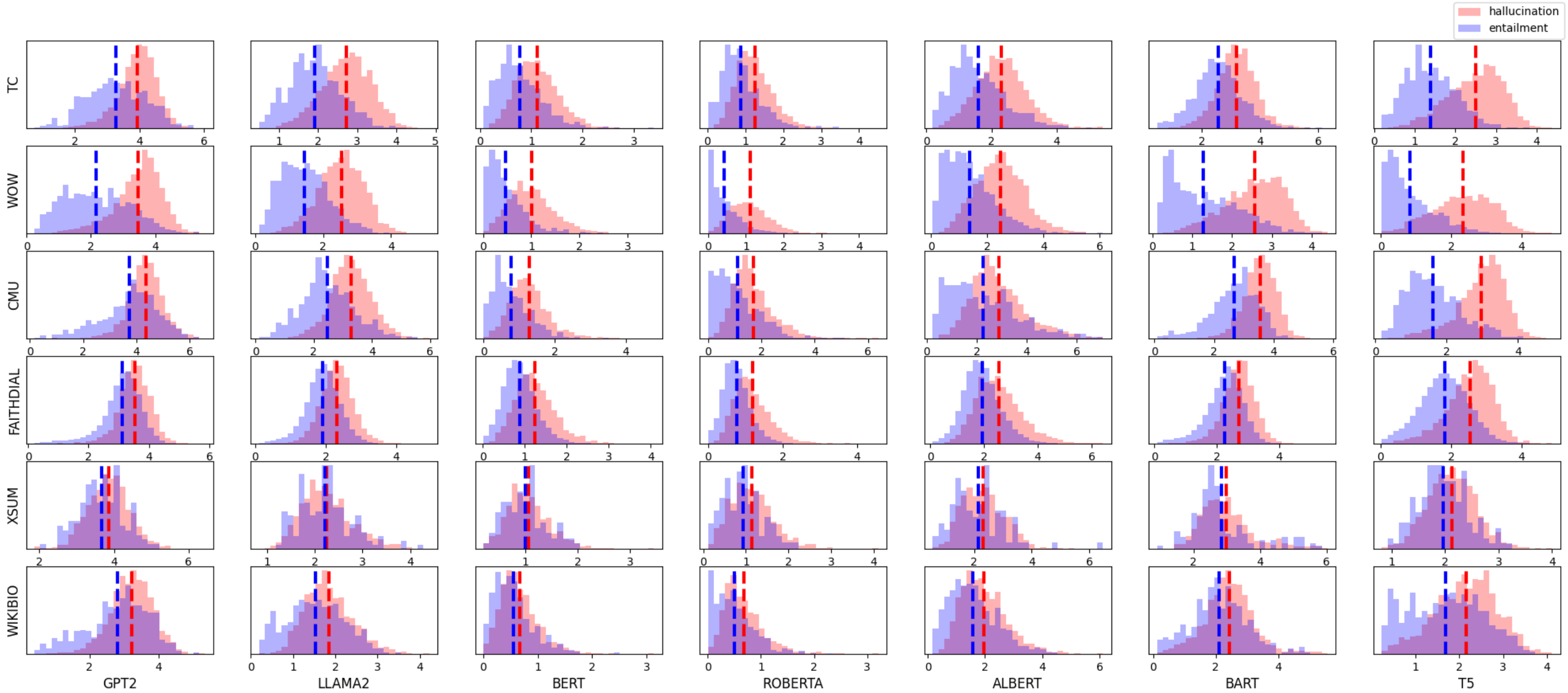
- We compute two metrics
  - Generation Probability (LogProb)
  - Entropy
- With 24 different sizes and types of PLMs
  - Encoder (BERT, RoBERTa, ALBERT)
  - Decoder (GPT2, LLAMA2)
  - Encoder-Decoder (T5, BART)
- On 6 data sets
  - Knowledge-grounded dialogue (TC, WOW, CMU)
  - Summarization (XSUM)
  - Wiki-like generation (WikiBio)

- Compare with two statistics
  - Kolmogorov-Smirnov statistics
  - Wasserstein distance



AIML@K KOREA UNIVERSITY

Distinguishability

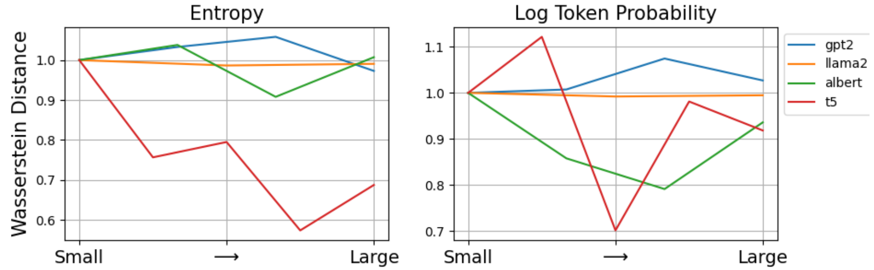


- Regardless of model size and type, 88-98% of cases return statistically significantly distinguishable distributions
- Both LogProb and Entropy show significant distinguishability

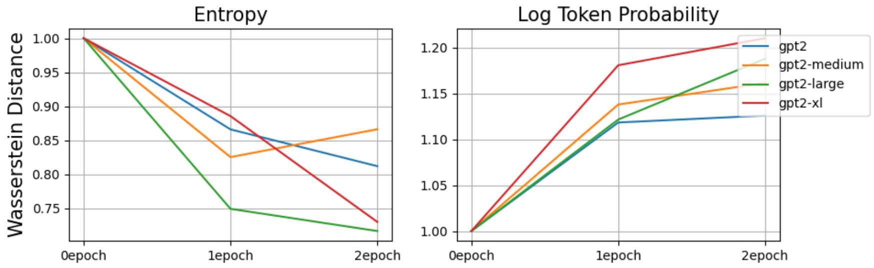
AIML@K KOREA UNIVERSITY

Distinguishability

- Size Effect
  - Researchers tend to adopt the largest (like GPT-4) model first
  - But bigger size does not guarantee better distinguishability



- Training Effect
  - Several hallucination-reduction methods utilize trained models
  - But training affect distinguishability in both (un)favorable ways



AIML@K KOREA UNIVERSITY

Weighted Training

- We observed the hallucinated data points show higher Entropy and lower LogProb.

- We propose a training method with the loss weighted by LogProb and Entropy of each data point.

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 $M \in \{\text{Entropy}, \text{LogProb}\}$, 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 $w_i \text{Loss}(x_i, y_i)$

Weighted Training

- Compare with 4 baselines
  - Unweighted: Usual training
  - CTRL: Control-token based method for knowledge grounded dialogue
  - Truncation: Truncate high-loss data points for summarization
  - mFACT: Weight loss with faithfulness score for summarization

- On 3 data sets
  - WOW, FaithDial: Knowledge grounded dialogue data sets
  - MediQA: Summarization + QA data set

- With 4 faithfulness metrics and 3 general text quality measures

Weighted Training

Data Set	Method	Q^2		SummaC	FactKB	ROUGE-L	BERT Score	BART Score
		F1	NLI					
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)

- Our method improves faithfulness while maintaining overall text quality.
- It also shows general applicability through various tasks.

Key Takeaways

- PLMs generally return distinguishable distributions to unfaithfully hallucinated texts
- NLPers should check the size and training effect before adopting the models
- We derive a simple but effective training method which enhance the model faithfulness

Paper

Code