Transformer Language Models Handle Word Frequency in Prediction Head

Goro Kobayashi[1,3] **Tatsuki Kuribayashi**[2,1] **Sho Yokoi**[1,3] **Kentaro Inui**[1,3]
[1] Tohoku University [2] MBZUAI [3] RIKEN
goro.koba@dc.tohoku.ac.jp tatsuki.kuribayashi@mbzuai.ac.ae
{yokoi, kentaro.inui}@tohoku.ac.jp

Abstract

Prediction head is a crucial component of Transformer language models. Despite its direct impact on prediction, this component has often been overlooked in analyzing Transformers. In this study, we investigate the inner workings of the prediction head, specifically focusing on bias parameters. Our experiments with BERT and GPT-2 models reveal that the biases in their word prediction heads play a significant role in the models' ability to reflect word frequency in a corpus, aligning with the logit adjustment method commonly used in long-tailed learning. We also quantify the effect of controlling the biases in practical auto-regressive text generation scenarios; under a particular setting, more diverse text can be generated without compromising text quality.

https://github.com/gorokoba560/transformer-lm-word-freq-bias

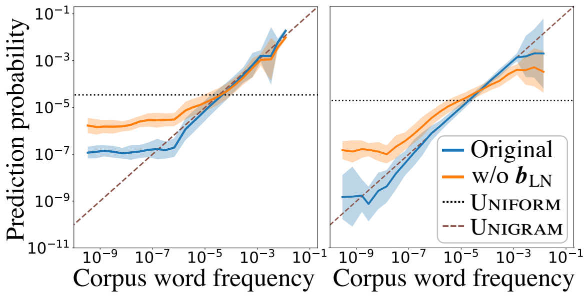

Figure 1: Changes in word prediction probabilities due to the removal of bias $\boldsymbol{b}_{\mathrm{LN}}$ from BERT base (left) and GPT-2 small (right).

1 Introduction

Transformer language models (TLMs) (Devlin et al., 2019; Radford et al., 2019) are now fundamental to natural language processing (NLP) techniques, including text generation. Owing to this success, extensive research has been conducted to analyze their inner workings (Rogers et al., 2020; Geva et al., 2021).

In this study, we shed light on the operation of the **prediction head**, the last block of the TLMs. Despite its direct impact on TLMs' output, its characteristics have been overlooked in previous analyses. Our experiments with BERT and GPT-2 reveal that **a particular bias parameter in the prediction head adjusts the model's output toward word frequency in a corpus**. Particularly, the bias increases the prediction probability for high-frequency words and vice versa (Figure 1).

We further explore this phenomenon from several perspectives. First, we analyze the geometric characteristics of this phenomenon, which show that word frequency is encoded in a specific direction in the output embedding space. Second, we analyze the behavioral impact of controlling their frequency biases on text generation. The results demonstrate that the model's text generation can be made more diverse while maintaining the fluency by adequately decaying the bias parameters, suggesting that models can more or less isolate word frequency knowledge from other text generation ability. Third, we discuss the potential connection between our findings and the logit adjustment method that is typically used in the machine learning field to address the class imbalance problem.

2 Background: Prediction Head

TLMs have a stack of Transformer layers on top of the embedding layer. These components update hidden token representations (Figure 2). **Prediction head**, which is our target of analysis, is the last, top-most component in TLMs. The prediction head in a TLM computes the prediction probabilities for all vocabulary $\mathcal{V}$ based on the hidden state in the last Transformer layer.

Formally, the prediction head receives, for each token, the last Transformer layer's hidden state $\boldsymbol{x} \in \mathbb{R}^d$. The prediction head computes the probability distribution $\boldsymbol{p} \in \mathbb{R}^{|\mathcal{V}|}$ of the next word as follows,

Findings of the Association for Computational Linguistics: ACL 2023, pages 4523–4535
July 9-14, 2023 ©2023 Association for Computational Linguistics

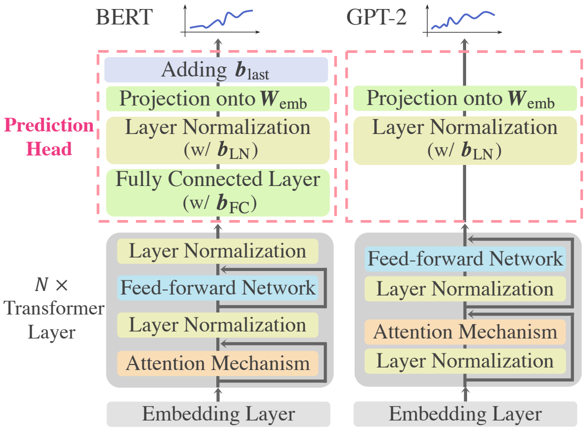

Figure 2: Architecture overview of BERT and GPT-2.

in the case of GPT-2:

$$\boldsymbol{p} = \mathrm{softmax}\Big(\mathrm{LN}(\boldsymbol{x})\boldsymbol{W}_{\text{emb}}\Big) \tag{1}$$

$$\mathrm{LN}(\boldsymbol{x}) := \frac{\boldsymbol{x} - m(\boldsymbol{x})}{s(\boldsymbol{x})} \odot \boldsymbol{\gamma} + \boldsymbol{b}_{\text{LN}} \in \mathbb{R}^d, \tag{2}$$

where $\boldsymbol{W}_{\text{emb}} \in \mathbb{R}^{d\times|\mathcal{V}|}$ denotes the word embedding matrix, $m(\boldsymbol{x})$ and $s(\boldsymbol{x})$ denote the element-wise mean and standard deviation, respectively, and $\odot$ denotes the element-wise product. $\boldsymbol{\gamma}$ and $\boldsymbol{b}_{\text{LN}} \in \mathbb{R}^d$ are learnable parameters.

For BERT, there is an additional fully connected layer (FC). The prediction head computes the probability distribution $\boldsymbol{p}$ for the hidden state $\boldsymbol{x}$ that corresponds to the [MASK] token as follows:

$$\boldsymbol{p} = \mathrm{softmax}\Big(\mathrm{LN}(\boldsymbol{x}')\boldsymbol{W}_{\text{emb}} + \boldsymbol{b}_{\text{last}}\Big) \tag{3}$$

$$\boldsymbol{x}' = \mathrm{GELU}\big(\boldsymbol{x}\boldsymbol{W}_{\text{FC}} + \boldsymbol{b}_{\text{FC}}\big) \in \mathbb{R}^d, \tag{4}$$

where $\boldsymbol{W}_{\text{FC}} \in \mathbb{R}^{d\times d}$ denotes the learnable weight matrix, and $\boldsymbol{b}_{\text{FC}} \in \mathbb{R}^d$ and $\boldsymbol{b}_{\text{last}} \in \mathbb{R}^{|\mathcal{V}|}$ denote the learnable bias parameters. GELU (Hendrycks and Gimpel, 2016) is the activation function.

Both prediction heads contain the bias $\boldsymbol{b}_{\text{LN}}$, and the BERT head additionally contains the biases $\boldsymbol{b}_{\text{FC}}$ and $\boldsymbol{b}_{\text{last}}$. As the first step in analyzing the prediction head, we focus on these three biases because they can easily be mapped to the output space. Drawing on the existing findings about the frequency-related workings of several components in the Transformer (Voita et al., 2019; Kobayashi et al., 2020), we analyze the model behavior with respect to word frequency.

3 Experiments

First, we show that the bias parameters are related to word frequency. Next, we analyze their properties from two perspectives: (i) geometric characteristics and (ii) text generation.

Model: We used BERT (cased) (Devlin et al., 2019) in two different sizes (base and large) and GPT-2 (Radford et al., 2019) in four different sizes (small, medium, large, and xl).

Data: We used 5,000 sequences from the test set of the GPT-2 pre-trainng corpus, OpenWebText Corpus (Gokaslan and Cohen, 2019)[1]. Each sequence was fed into BERT after some tokens were replaced with [MASK][2], and fed into GPT-2 as it was. Further, word frequencies were calculated from the corpus used for training each of BERT and GPT-2.[3]

3.1 Impact of biases on prediction distribution

We compared the models' word prediction with and without each bias. Specifically, we once obtained word prediction distributions $\hat{\boldsymbol{p}} \in \mathbb{R}^{|\mathcal{V}|}$ from a model for each time step across the test data. The average of these distributions are referred to as *model's word prediction distribution* henceforth.

Bias adjusts the model's prediction distribution closer to the corpus frequency distribution: Figure 1 shows changes in the model's word prediction distribution before and after the bias $\boldsymbol{b}_{\text{LN}}$ is removed.[4] The removal of $\boldsymbol{b}_{\text{LN}}$ increases the probability of the model predicting low-frequency words (right side of the figures) and vice versa, which results in a word prediction distribution that approaches a flat (UNIFORM in the figure). In other words, the bias $\boldsymbol{b}_{\text{LN}}$ adjusts the models' word prediction distribution to be closer to the corpus word frequency distribution (UNIGRAM in the figure). This finding can be generalized across all model sizes (Appendix A).

To quantify the above effect, we calculated the Kullback–Leibler (KL) divergence between the model's word prediction distribution and the corpus word frequency distribution (UNIGRAM). Note that a *higher* value indicates that the model's prediction distribution has *more discrepancy* with that in

[1] `webtext.test.jsonl` published in https://github.com/openai/gpt-2-output-dataset was used.

[2] Following Devlin et al. (2019), 15% of tokens were replaced with [MASK] 80% of the time.

[3] BERT was trained on Wikipedia and BooksCorpus (Zhu et al., 2015), and GPT-2 was trained on OpenWebText Corpus. We reproduced them using `Datasets` (Lhoest et al., 2021).

[4] We created bins to divide the corpus word frequencies into constant intervals and plotted each bin's geometric mean and standard deviation for the word prediction probabilities.

Model		Original	w/o $\boldsymbol{b}_{\mathrm{LN}}$	w/o $\boldsymbol{b}_{\mathrm{FC}}$	w/o $\boldsymbol{b}_{\mathrm{last}}$
BERT	base	0.20	**0.39**	0.22	0.23
	large	0.21	**0.39**	0.23	0.23
GPT-2	small	0.14	**0.83**	-	-
	medium	0.14	**0.34**	-	-
	large	0.14	**0.17**	-	-
	xl	0.14	**0.17**	-	-

Table 1: KL divergence between the model's word prediction distribution and the corpus word frequency distribution. A larger value means that the distributions are more divergent. $\boldsymbol{b}_{\mathrm{FC}}$ and $\boldsymbol{b}_{\mathrm{bias}}$ are contained only in BERT.

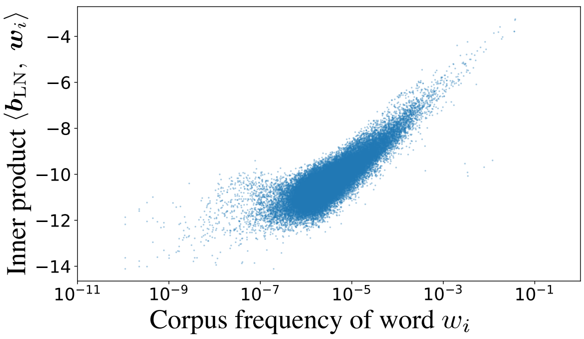

Figure 3: Relationship between the corpus word frequency and inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in GPT-2 small.

the pretraining corpus. Table 1 shows that removing $\boldsymbol{b}_{\mathrm{LN}}$ always results in a higher value, which indicates that $\boldsymbol{b}_{\mathrm{LN}}$ indeed adjusts the prediction distribution to be closer to the corpus frequency distribution. The biases $\boldsymbol{b}_{\mathrm{FC}}$ and $\boldsymbol{b}_{\mathrm{last}}$ in BERT also exert a similar effect, but it is weaker than that of $\boldsymbol{b}_{\mathrm{LN}}$; we focus on $\boldsymbol{b}_{\mathrm{LN}}$ in the following. We also observe that larger models have less change of frequency biases due to $\boldsymbol{b}_{\mathrm{LN}}$.

3.2 Geometric observations

We observed the geometric properties of the bias $\boldsymbol{b}_{\mathrm{LN}}$ and the output embedding space of the TLMs.

Word frequency is encoded in the bias vector's direction in the output embedding space:

The observation that the bias vector shifts predictions according to word frequency suggests that word frequency is encoded in the output embedding space $\boldsymbol{W}_{\mathrm{emb}}$, and the bias vector $\boldsymbol{b}_{\mathrm{LN}}$ is a good projection to extract this frequency information. In fact, the inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each word embedding $\boldsymbol{w}_i$ in the embedding layer correlates well with the word frequency[5] (Figure 3).

Furthermore, we observed that removing the bias direction ($\approx$ frequency direction) from the embedding matrix $\boldsymbol{W}_{\mathrm{emb}}$ improved the isotropy (uniformity in direction, e.g., Ethayarajh, 2019) in the output embedding space. Formally, we removed the bias direction using $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i - \langle \boldsymbol{w}_i, \frac{\boldsymbol{b}_{\mathrm{LN}}}{\|\boldsymbol{b}_{\mathrm{LN}}\|} \rangle \frac{\boldsymbol{b}_{\mathrm{LN}}}{\|\boldsymbol{b}_{\mathrm{LN}}\|}$; then, the average value $\frac{1}{n^2}\sum_i \sum_j \cos(\boldsymbol{w}_i, \boldsymbol{w}_j)$ decreased from 0.15 to 0.09 in BERT base. This observation shows that the anisotropy in the output space is, more or less, caused by the frequency direction.

We further observed that hidden states $\boldsymbol{h}_{\mathrm{token}}$ before $\boldsymbol{b}_{\mathrm{LN}}$ was added were almost orthogonal to $\boldsymbol{b}_{\mathrm{LN}}$ ($\approx$ word frequency direction); in particular, $\mathbb{E}_{\mathrm{token}} \left|\cos\left(\boldsymbol{h}_{\mathrm{token}},\ \boldsymbol{b}_{\mathrm{LN}}\right)\right| = 0.08 \ll 1.0$ in BERT-base. This corroborates that the frequency bias injected in the prediction head indeed does not exist in the hidden states before the prediction head.

Word frequency encoded on the bias vector is shifted via fine-tuning:

We also inspected whether the model's word prediction distribution is shifted to that in the target domain during fine-tuning to enhance the generality of our observation. Specifically, we fine-tuned GPT-2 small on a dataset consisting of abstracts from papers in the machine learning field[6], whose word frequency distribution is different from the pretraining data. After fine-tuning, the inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each word embedding $\boldsymbol{w}_i$ correlated more with the additional fine-tuning corpus after fine-tuning (the Spearman's ρ changed from 0.38 to 0.62) and slightly less with the pre-training corpus (the Spearman's ρ changed from 0.78 to 0.73). This suggests that frequency information captured by the bias $\boldsymbol{b}_{\mathrm{LN}}$ is updated during fine-tuning.

3.3 Impact of bias on text generation

We next demonstrate that controlling the bias $\boldsymbol{b}_{\mathrm{LN}}$ can lead to more diverse text generation without significant harm to the quality of the text. We hope that quantifying the effect of such control using metrics for the evaluation of text generation (e.g., n-gram diversity) will enhance the connection between the language generation field and the field of probing/interpreting LMs' internals.

[5] Spearman's ρ was 0.78 on GPT-2 small.

[6] `CShorten/ML-ArXiv-Papers` published in https://huggingface.co/datasets/CShorten/ML-ArXiv-Papers on `Datasets` (Lhoest et al., 2021) was used.

Procedure: We adjusted $\boldsymbol{b}_{\text{LN}}$ during text generation by GPT-2, and we then evaluated the generated text. Specifically, we introduced an adjustment coefficient $\lambda \in [0, 1]$ and replaced $\boldsymbol{b}_{\text{LN}}$ with $\lambda \boldsymbol{b}_{\text{LN}}$. We report the evaluation scores by varying λ. The results generated with the top-p sampling strategy (Fan et al., 2018) are reported in this section. The results for other decoding settings are in Appendix A; we found similar results for the top-p and top-k sampling but found degradation with the vanilla sampling setting. The details of the settings are described in Appendix B.

Evaluation methods: Text generated by each model was evaluated from two perspectives: diversity and quality. For the diversity evaluation, Distinct-n D_n (Li et al., 2016) and N-gram diversity D (Meister et al., 2022) were used. These measures of n-gram overlap in generated texts were calculated as follows:

$$D_n(\text{texts}) := \frac{\#\ \text{Unique } n\text{-grams in texts}}{\#\ n\text{-grams in texts}} \tag{5}$$

$$D(\text{texts}) := \frac{1}{4} \sum_{n=1}^{4} D_n(\text{texts}). \tag{6}$$

For the quality evaluation, MAUVE (Pillutla et al., 2021) and Perplexity (PPL) were used. MAUVE evaluates how similar a given text generation model is to humans by comparing human-written texts and model-generated texts according to the difference in their distributions in a sentence embedding space. PPL evaluates how well models can predict words in human-written texts.

Results: Table 2 shows the results. Weakening the bias $\boldsymbol{b}_{\text{LN}}$ ($\lambda < 1$) increased the diversity of the generated text but decreased the PPL score, exhibiting a general trade-off between them. Nevertheless, for the larger models, GPT-2 large ($\lambda = 0.5$) and xl ($\lambda = 0.7$), there was a sweet spot, where the diversity and the MAUVE score improved with little decrease in PPL. This observation can be interpreted as follows. The larger models were able to predict the context-dependent probability of low-frequency words as precisely as that of high-frequency words, so promoting low-frequency words with those models improved the diversity while maintaining the quality of the text. The smaller models were equally accurate in predicting the probability of high-frequency words but tended to be inaccurate for low-frequency words, so promoting low-frequency words degraded the quality of the text. This interpretation is also consistent with the class imbalance problem, which will be discussed in Section 4.1. From the application perspective, this observation also suggests that the lexical diversity in text generation can be improved simply by modifying particular parameters in the prediction head.

Model	λ	Diversity ↑			Quality	
		D_1	D_2	D	MAUVE ↑	PPL ↓
small	1	0.04	0.32	0.49	**0.85**	**19.4**
	0.6	**0.06**	**0.61**	0.59	0.18	24.1
	0	0.04	0.36	0.32	0.01	65.9
med.	1	0.05	0.35	0.51	0.90	**14.6**
	0.9	0.05	0.39	0.54	**0.90**	14.8
	0.2	0.07	**0.63**	0.60	0.14	18.8
	0	**0.08**	0.60	0.55	0.06	21.3
large	1	0.04	0.30	0.47	0.90	**12.7**
	0.5	0.04	0.36	0.50	**0.91**	12.9
	0	**0.04**	**0.42**	**0.54**	0.86	13.6
xl	1	0.04	0.30	0.47	0.90	**11.4**
	0.7	0.04	0.34	0.49	**0.92**	11.5
	0	**0.04**	**0.41**	**0.53**	0.86	12.1

Table 2: Evaluation results for GPT-2 (top-p sampling) while bias $\boldsymbol{b}_{\text{LN}}$ was controlled with λ. Results for $\lambda = 0, 1$, and other notable values are listed.

We also show several samples of the generated texts (Appendix C). We generally observed that overly decreasing λ incurs (i) more proper nouns, (ii) more repetitions of the same words or similar phrases, and (iii) the generation of ungrammatical text, especially for the small models. This may also be related to the suppression of the punctuation and end-of-sequence token, which are highly frequent.

4 Discussion

4.1 Connection with logit adjustment methods

We revealed that adding the bias $\boldsymbol{b}_{\text{LN}}$ (which was performed immediately before the logit was computed) encourages TLMs to generate high-frequency words, and de-biasing promotes diversity. This can also be seen as analogous to logit adjustment, which is a common technique for addressing the class imbalance problem, where the label (the word in text generation) frequency distribution is long-tailed (Provost, 2000; Zhou and Liu, 2006; Collell et al., 2016; Menon et al., 2021). In particular, Menon et al. (2021) proposed to minimize the balanced error (i.e., an average of per-class errors) by directly adding the label frequency distribution to logits during training but not during inference.

One can find an analogy between the modification of $\boldsymbol{b}_{\mathrm{LN}}$ and their method: (i) adding the frequency-shifting bias $\boldsymbol{b}_{\mathrm{LN}}$ corresponds to the operation of adding the class-frequency-based margins to the logits; (ii) promoting low-frequency words by removing $\boldsymbol{b}_{\mathrm{LN}}$ during inference corresponds to the way logit adjustment encourages low-class prediction. In other words, interestingly, TLMs seem to implicitly learn something similar to balanced error minimization without being explicitly designed to do so (e.g., loss function).

4.2 Connection with a technique to initialize bias parameter with class frequency

In training neural classification models, using class frequency to initialize the last bias to be added to the logit is a well-known and efficient technique (Karpathy, 2019). Therefore, our observation that the bias vector at the prediction head (i.e., the last block) encodes word frequency might seem somewhat obvious. However, our experimental results showed peculiar trends that might be stemmed from the inductive bias of TLMs. First, although the initialization technique implies the relationship between the *last* bias $\boldsymbol{b}_{\mathrm{last}}$ and the corpus word frequency, we found that the bias $\boldsymbol{b}_{\mathrm{LN}} \in \mathbb{R}^d$, which is further away from the output and less expressive than $\boldsymbol{b}_{\mathrm{last}} \in \mathbb{R}^{|\mathcal{V}|}$, plays the role in encoding the frequency in BERT (Table 1). For GPT-2, not even $\boldsymbol{b}_{\mathrm{last}}$ exists. Second, even $\boldsymbol{b}_{\mathrm{LN}}$ plays a weak role in encoding the frequency in larger models (Table 1). These findings suggest that neural models dynamically determines the role of each internal module according to various factors such as parameter size and architecture. When and under what conditions the short vector $\boldsymbol{b}_{\mathrm{LN}}$ strongly encodes the frequency is an interesting question and left to future research.

5 Related work

Transformer layers (e.g., attention patterns) have been the major focus of TLM analysis (Clark et al., 2019; Mareček and Rosa, 2019; Kobayashi et al., 2021; Dai et al., 2022). The first embedding layer, especially positional encoding, has also been studied (Wang et al., 2021; Kiyono et al., 2021). This study sheds light on the prediction head, the last block of a TLM, and provides new insights into the working mechanisms of TLMs.

Notably, previous studies have reported that words having a similar frequency are clustered in the embedding spaces of various deep NLP models (Mu and Viswanath, 2018; Gong et al., 2018; Provilkov et al., 2020; Liang et al., 2021); our observation agrees with theirs. In addition to this, we newly discovered that a particular bias parameter in the TLM prediction head corresponds to “word frequency direction” in the word embedding space.

6 Conclusions

In this study, we explored the workings of bias parameters in the prediction head of TLMs. Our experiments with BERT and GPT-2 showed that the biases adjust the model’s prediction with respect to word frequency. We further explored this phenomenon and provided the following insights: (i) word frequency is encoded in a specific direction (the bias direction) in the output embedding space, (ii) properly controlling the bias’s effect can encourage more diverse language generation without compromising quality, and (iii) TLMs are implicitly trained to be potentially consistent with the logit adjustment method. In future work, we will analyze larger TLMs, e.g., Open Pre-trained Transformers (Zhang et al., 2022). Further, we will analyze the weight parameters in the prediction head in addition to the bias parameters.

Limitations

There are mainly two limitations in this study. First, we still do not consider components other than the bias parameters in the prediction head. For example, the weight parameters of the prediction head, i.e., γ and $\boldsymbol{W}_{\mathrm{FC}}$, can also affect a model’s prediction. Second, our findings do not cover the Transformer language models other than BERT (base and large) and GPT-2 (small, medium, large, and xl). Consistent findings were obtained for the two main architectures (i.e., encoder-based masked, and decoder-based causal language models) and for various model sizes, although future research is needed to show whether the findings can be generalized to RoBERTa (Liu et al., 2019), Open Pre-trained Transformer Language Models (OPT, Zhang et al., 2022), and other variants. Considering Transformer encoder-decoder models, such as neural machine translation models and T5 (Raffel et al., 2020), would also be an interesting future direction.

Ethics Statement

This paper sheds light on the workings of the prediction head of the fundamental models in NLP. In recent years, unintended biases (e.g., gender bias) in neural network models have been problematic. This paper may help in this direction by encouraging researchers to analyze the prediction head as well as Transformer layers.

Acknowledgements

We would like to thank the members of the Tohoku NLP Group for their insightful comments, particularly Shiki Sato for his valuable suggestions regarding experimental settings. This work was supported by JSPS KAKENHI Grant Number JP22J21492, JP22H05106; JST CREST Grant Number JPMJCR20D2, Japan; and JST ACT-X Grant Number JPMJAX200S, Japan.

References

Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Guillem Collell, Drazen Prelec, and Kaustubh Patil. 2016. Reviving Threshold-Moving: a Simple Plug-in Bagging Ensemble for Binary and Multiclass Imbalanced Data. *arXiv preprint 1606.08698v3*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8493–8502.

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 (NAACL-HLT)*, pages 4171–4186.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 889–898.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. FRAGE: Frequency-Agnostic Word Representation. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 1341–1352.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). *arXiv preprint 1606.08415v4*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations (ICLR)*.

Andrej Karpathy. 2019. A Recipe for Training Neural Networks. Andrej Karpathy blog.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted Absolute Position Embedding for Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3309–3321.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4568.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing (EMNLP): System Demonstrations, pages 175–184.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 110–119.

Yuxin Liang, Rui Cao, Jie Zheng, Jie Ren, and Ling Gao. 2021. Learning to Remove: Towards Isotropic Pre-trained BERT Embedding. In *Artificial Neural Networks and Machine Learning (ICANN)*, pages 448–459.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint*, cs.CL/1907.11692v1.

David Mareček and Rudolf Rosa. 2019. From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. Locally Typical Sampling. *arXiv preprint 2202.00666v4*.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *9th International Conference on Learning Representations (ICLR)*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *6th International Conference on Learning Representations (ICLR)*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1882–1892.

Foster Provost. 2000. Machine Learning from Imbalanced Data Sets 101. In *Proceedings of the AAAI 2000 Workshop on Imbalanced Data Sets*, pages 1–3.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

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 (JMLR)*, 21(140):1–67.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics (TACL)*, 8:842–866.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406.

Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. 2021. On Position Embeddings in BERT. In *9th International Conference on Learning Representations (ICLR)*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *arXiv preprint 2205.01068v4*.

Zhi-Hua Zhou and Xu-Ying Liu. 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Experimental results in other settings

In Section 3.1, we presented the changes in the word prediction distribution before and after removing the bias $\boldsymbol{b}_{\mathrm{LN}}$ of BERT base and GPT-2 small in Figure 1. The results of the other models are shown in Figures 4 to 7.

In Section 3.2, we showed that the inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each output word embedding $\boldsymbol{w}_i$ correlated well with the word frequency for GPT-2 small (Figure 3). The results for the other models are shown in Figures 8 to 12. The Spearman's correlation coefficient is listed in Table 3.

In Section 3.3, we showed the effect of controlling the bias $\boldsymbol{b}_{\text{LN}}$ on GPT-2's text generation with a top-p sampling strategy. We also conducted experiments with other sampling strategies: top-k sampling (Holtzman et al., 2020) and vanilla sampling. The results of these two sampling strategies are listed in Tables 4 and 5. We found that the results of top-k sampling were similar to those of top-p sampling; for the larger models, GPT-2 large ($\lambda = 0.3$) and xl ($\lambda = 0.5$), there also was a sweet spot, where diversity and MAUVE improved with little decrease in PPL. In contrast, with vanilla sampling, both MAUVE and PPL decreased consistently and quickly.

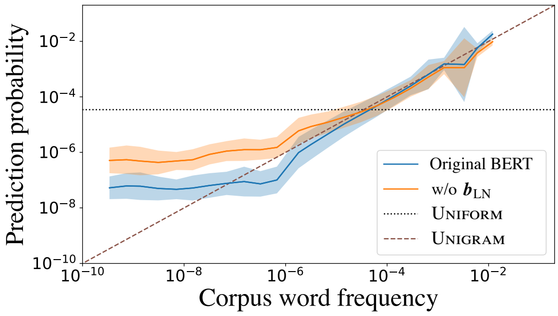

Figure 4: Changes in word prediction probabilities due to bias $\boldsymbol{b}_{\text{LN}}$ removal on BERT large.

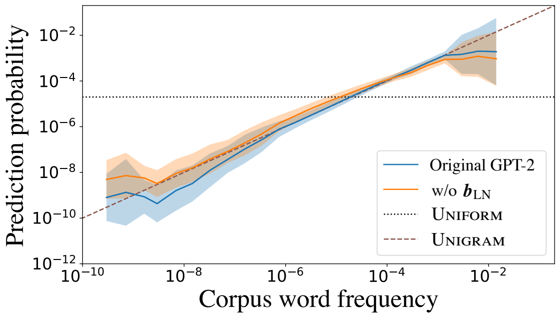

Figure 5: Changes in word prediction probabilities due to bias $\boldsymbol{b}_{\text{LN}}$ removal on GPT-2 medium.

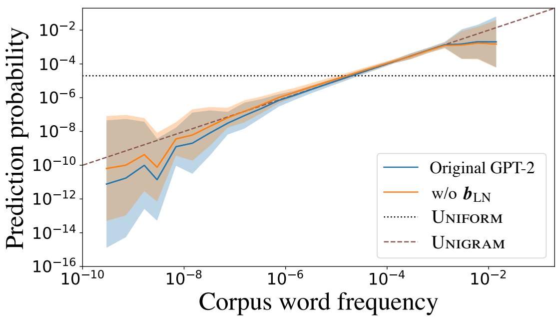

Figure 6: Changes in word prediction probabilities due to bias $\boldsymbol{b}_{\text{LN}}$ removal on GPT-2 large.

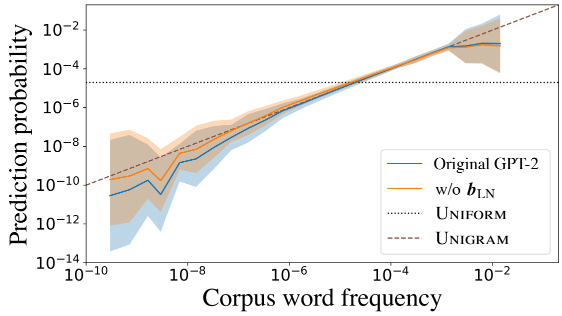

Figure 7: Changes in word prediction probabilities due to bias $\boldsymbol{b}_{\text{LN}}$ removal on GPT-2 xl.

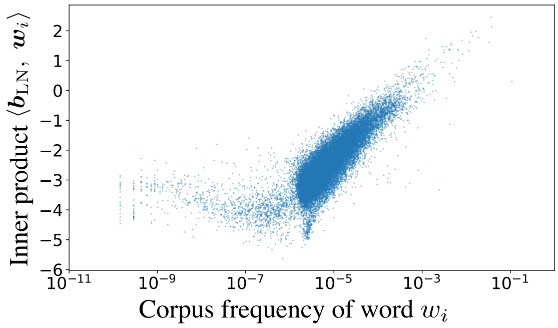

Figure 8: Relationship between the corpus word frequency and the inner product of $\boldsymbol{b}_{\text{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in BERT base.

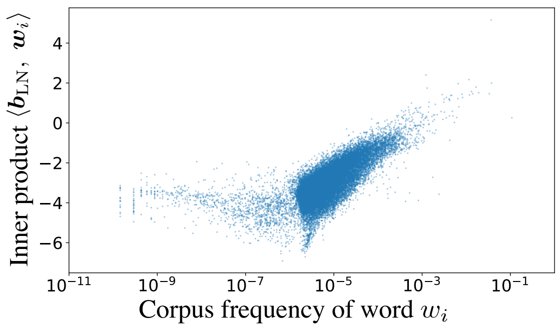

Figure 9: Relationship between the corpus word frequency and the inner product of $\boldsymbol{b}_{\text{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in BERT large.

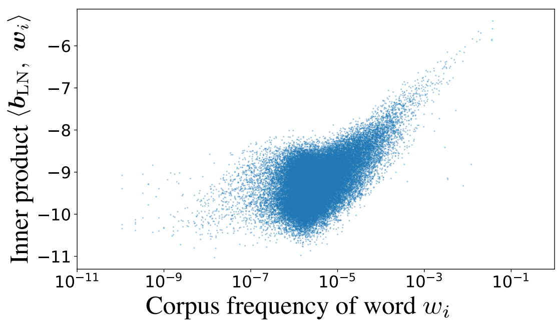

Figure 10: Relationship between the corpus word frequency and the inner product of $\boldsymbol{b}_{\text{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in GPT-2 medium.

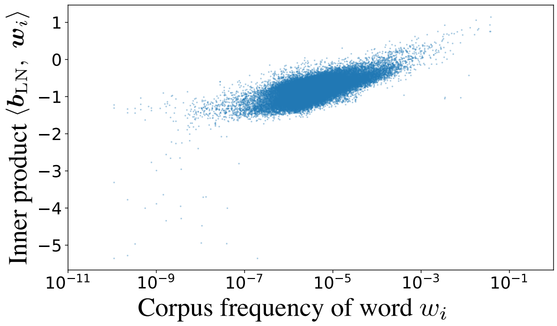

Figure 11: Relationship between the corpus word frequency and the inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in GPT-2 large.

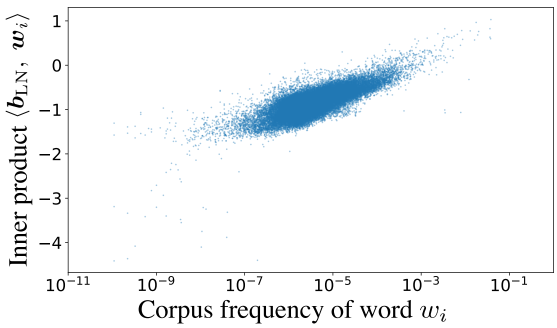

Figure 12: Relationship between the corpus word frequency and the inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each output word embedding $\boldsymbol{w}_i$ in GPT-2 xl.

Model		Spearman's ρ
BERT	base	0.84
	large	0.74
GPT-2	small	0.78
	medium	0.43
	large	0.61
	xl	0.70

Table 3: The Spearman's correlation coefficient between the corpus word frequency and inner product of $\boldsymbol{b}_{\mathrm{LN}}$ and each output word embedding $\boldsymbol{w}_i$.

Model	λ	Diversity ↑			Quality	
		D_1	D_2	D	MAUVE ↑	PPL ↓
small	1	0.03	0.23	0.42	0.78	**19.4**
	0.9	0.03	0.27	0.45	**0.82**	19.8
	0.7	**0.03**	**0.34**	**0.48**	0.72	22.0
	0	0.02	0.12	0.13	0.01	65.9
med.	1	0.03	0.27	0.46	**0.89**	**14.6**
	0.3	0.03	**0.38**	0.50	0.64	17.8
	0	0.03	0.33	0.46	0.22	21.3
large	1	0.03	0.26	0.44	0.89	**12.7**
	0.3	**0.03**	0.32	0.48	**0.90**	13.1
	0	0.03	**0.34**	**0.50**	0.87	13.6
xl	1	0.03	0.28	0.45	0.92	**11.4**
	0.5	0.03	0.32	0.48	**0.92**	11.6
	0	**0.03**	**0.36**	**0.50**	0.89	12.1

Table 4: Evaluation results for GPT-2 (top-k sampling) while bias $\boldsymbol{b}_{\mathrm{LN}}$ was controlled with λ.

Model	λ	Diversity ↑			Quality	
		D_1	D_2	D	MAUVE ↑	PPL ↓
small	1	0.07	0.49	0.59	**0.50**	**19.4**
	0.5	**0.14**	**0.88**	0.73	0.02	27.0
	0	0.12	0.71	0.61	0.01	65.9
med.	1	0.09	0.56	0.63	**0.33**	**14.6**
	0.2	0.19	**0.86**	**0.74**	0.03	18.8
	0	**0.21**	0.86	0.74	0.02	21.3
large	1	0.06	0.44	0.56	**0.77**	**12.7**
	0.5	0.08	0.55	0.61	0.53	12.9
	0	**0.11**	**0.69**	**0.67**	0.22	13.6
xl	1	0.06	0.43	0.56	**0.82**	**11.4**
	0.5	0.08	0.54	0.61	0.61	11.6
	0	**0.11**	**0.68**	**0.67**	0.24	12.1

Table 5: Evaluation results for GPT-2 (vanilla sampling) while bias $\boldsymbol{b}_{\mathrm{LN}}$ was controlled with λ.

B Detailed experimental settings

To observe the TLM word prediction distribution (the main experiments in Section 3.1 and the measurement of PPL in Section 3.3), we let BERT predict words corresponding to `[MASK]` tokens, and we let GPT-2 predict the second and subsequent words in each sequence. If the length of an input sequence was greater than the maximum input length k of the model, only the first k words were used.

To evaluate the TLM text generation (Section 3.3), the first 10 words of each sequence were fed into to GPT-2, and subsequent words were generated until the length of the sequence reached 1,024 words or the end-of-sequence token was generated. For GPT-2 small and medium, we varied λ in increments of 0.1 to control the bias $\boldsymbol{b}_{\mathrm{LN}}$. For GPT-2 large and xl, we first checked the results for 100 samples and obtained the values with some kind of trends; we then varied λ in $\{0, 0.3, 0.5, 0.7, 1.0\}$ for the entire dataset, including the values.

We experimented with three decoding strategies: vanilla sampling, top-k sampling, and top-p sampling. In the top-k sampling, k was set to 50. In the top-p sampling, p was set to 0.9. Furthermore, before we evaluated the model-generated texts with the N-gram based diversity metrics, we applied the word tokenizer provided by NLTK (Bird and Loper, 2004).

C Examples of generated text

Table 6 shows examples of text generated by GPT-2 small and large while controlling the bias $\boldsymbol{b}_{\mathrm{LN}}$ with λ.

Model	λ	Generated text
small	1	There has been one product that I've wanted for a while — that is baseball's fountain and. I wanted to try to get another product to make it as polished and simple to use and even easier to push the right buttons. Have you played with some of the furniture brands of the past? Do you think the new smart building is going to...
	0.6	There has been one product that I've wanted for awhile: **Asus ZenUI** Keyboard Replacement Kit FAQ. I purchased this replacement keyboard replacement kit prior to 2014 when **Asus** shipped its **ZenUI** ... BIOS Reset Warranty Long warranty EUR 3500 EUR 4550 EUR 470 EUR 520 EUR 590 EUR 560 EUR...
	0	There has been one product that I've wanted for awhile got released that alters baseball's bench press. I mention **Alejandro Nazarovski** prior thus preferring **Julian Whitaker** however altering **Alejandro** ... combined with dumbbell movements combined with negatives ratios Improved athlete mobility Decreased fatigue Diseases Whilst adjusting lifts Underestimating injury Potential Extensions Suspension Period...
large	1	The **Atlanta Falcons** have started the 2015 season 4-0. (Photo: **Winslow Townson / Associated Press**) The Falcons' longest streak of consecutive seasons with a winning record started on the same day in Week 11 that **Mike Shanahan** and the Falcons experienced one of their most compelling victories of the season...
	0.5	The **Atlanta Falcons** have started the 2015 season 4-1, including a triumph over the **New Orleans Saints** at **Mercedes-Benz Stadium**. Look at what this team could be capable of as the season progresses. It has the goods, the direction, the talent to make a run at becoming a legitimate **Super Bowl** contender. More...
	0	The **Atlanta Falcons** have started the 2015 season 4-0, including a win over the **Minnesota Vikings** last Sunday night. It's been a perfect start to 2014 as well. Looking ahead, what's the road ahead? Week 1 @ **Tampa Bay Buccaneers** ...

Table 6: Examples of text generated by GPT-2 small and large with top-p sampling while bias $\boldsymbol{b}_{\mathrm{LN}}$ was controlled with λ. Proper nouns are in **bold**, repetitions of similar phrases are straight underlined, and ungrammatical passages are highlighted with wavy underlines. Note that the first 10 words are given to the model as context.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section after Conclusions (Section 5).

☑ A2. Did you discuss any potential risks of your work?
We discussed the generalizability of our findings in Limitations section after Conclusions (Section 5).

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction (Section 1) summarize our main claims.

☑ A4. Have you used AI writing assistants when working on this paper?
We used DeepL and Langsmith. DeepL is a machine translation tool, and Langsmith is a rephrasing tool trained on scientific text. We used them only to refine the English of our submission. Neither service is to copy the work of others nor make novel ideas or claims.

B ☑ Did you use or create scientific artifacts?

Section 3. We used published pre-trained models and corpus.

☑ B1. Did you cite the creators of artifacts you used?
Section 3.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We have already checked their licenses and the artifacts are enough popular not to need to discuss their license in the paper.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
It is obvious that all the artifacts we used were created in the context of the research and we also used them for the research purpose.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Since we used the dataset for evaluating models' workings, discussing about its specific content is not so essential for our paper. In addition, the dataset we used is the training corpus of the models. It is natural to use as is.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
All the artifacts we used are enough popular not to need to provide documentation.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 3.

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Since the purpose of our study is to analyze models' inner workings, the details of computational environment are not necessary.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3 and Appendix B.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.