Rare Word Representation Learning by Smoothing Over Word Classes

Anonymous ACL submission

Abstract

Language models strongly rely on frequency information because they maximize the likelihood of tokens during pre-training. As a consequence of this objective, language models tend to not generalize well to tokens rarely seen during training. Our work introduces a method for quantifying the **frequency bias** of a language model: the degree to which a language model is influenced by token frequency when determining the grammatical acceptability of sentences. We then present a method for pre-training a language model to remove the frequency bias by adjusting the objective function to distribute the learning signal to syntactically similar tokens, inducing a syntactic prior over the token embeddings. Our method, which we call **POS Smoothing**, results in better performance on infrequent tokens without degrading the model's general ability on downstream language understanding tasks. [1]

1 Introduction

Pre-trained transformer models have proven tremendously capable of solving a wide array of language understanding tasks (Touvron et al., 2023; Chowdhery et al., 2023). Part of the success of pre-trained language models can be attributed to the pre-training objective. Despite differences in architecture, the vast majority of language models are pre-trained to maximize the log-likelihood of a word, given the surrounding context (Devlin et al., 2019; Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). While the performance of language models has increased on a variety of language understanding benchmarks (Zellers et al., 2019; Hendrycks et al., 2020), it remains an open challenge to improve their performance on rare and specialized domains without resorting to perpetual increases of data and model size.

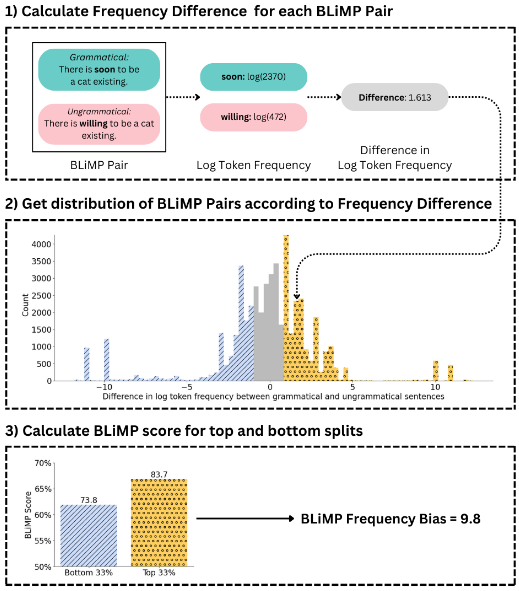
1) Calculate Frequency Difference for each BLiMP Pair
Grammatical: There is **soon** to be a cat existing.
Ungrammatical: There is **willing** to be a cat existing.
BLiMP Pair
soon: log(2370)
willing: log(472)
Log Token Frequency
Difference: 1.613
Difference in Log Token Frequency
2) Get distribution of BLiMP Pairs according to Frequency Difference
Count
4000
3500
3000
2500
2000
1500
1000
500
0
−10
−5
0
5
10
Difference in log token frequency between grammatical and ungrammatical sentences
3) Calculate BLiMP score for top and bottom splits
BLiMP Score
70%
65%
60%
55%
50%
73.8
83.7
Bottom 33%
Top 33%
BLiMP Frequency Bias = 9.8

Figure 1: Illustration of the BLiMP frequency bias calculation for our baseline model.

The fact that language follows a Zipfian distribution (Zipf, 1935) means that language models are exposed to frequent tokens exponentially more often than rare ones during pre-training. When focusing on cumulative evaluation scores, a language model's weak performance on low-frequency words is often overlooked.

To increase the generalization abilities of language models, it is essential to improve the performance on low-frequency tokens. Current approaches to modeling long tail distributions require large model sizes, limiting the scalability of these methods (Feldman, 2020; Haviv et al., 2023).

In this work, we propose **POS Smoothing**: a new method for improving representation learning in small language models. We smoothly distribute the backpropagation signal over syntactically similar tokens using a similarity metric based on part-of-speech tag distributions. Using this method, tokens that are rarely seen during training can fall back on a syntactically plausible initialization, taking on

[1] Our code is available (*CC BY-SA*): *anonymized*

representations of more frequent tokens that serve similar syntactic roles. We evaluate our method using a new metric for quantifying the frequency bias of language models (illustrated in fig. 1) and find that POS Smoothing improves the generalization capabilities of a small language model without affecting perplexity and downstream capabilities.

2 Representing Rare Tokens

Current approaches to language modeling rely on the memorization of infrequent words to perform well on generalization tasks (Feldman, 2020). However, models trained as likelihood maximizers tend to yield degenerate representations for rare words (Gao et al., 2019). To address this problem, Gong et al. (2018) introduces an adversarial training objective to push the embeddings of frequent and infrequent words to occupy a similar semantic space. Gao et al. (2019) proposes a novel regularization term to the standard log-likelihood objective to better distribute the representation of rare words in semantic space. Other approaches have shown promising results on rare word performance by constructing token embeddings that take into account a word's surface form as well as surrounding context (Schick and Schütze, 2019, 2020).

Memorization Effects Recent analytical work has shown that certain layers of transformer models implicitly encode frequency information (Kobayashi et al., 2023), while others store memorized long-tail data (Haviv et al., 2023). Feldman and Zhang (2020) demonstrate that models memorize atypical examples to achieve the highest accuracy on long-tailed data samples. This memorization hack, however, has only been shown to work well with over-parameterized models, when the number of weights surpasses the number of training samples (Belkin et al., 2019). These findings suggest that with the current limited training objectives, generalization can only be achieved by large models trained on noisy datasets with sufficient long-tail samples (Zheng and Jiang, 2022).

Integrating Linguistic Information In a separate line of work, researchers have explored the integration of morphological and orthographic information in word embeddings (Salle and Villavicencio, 2018; Vulić et al., 2017; Cotterell and Schütze, 2015; Bhatia et al., 2016; Botha and Blunsom, 2014). Others have proposed syntactically-motivated objective functions, such as predicting constituency labels (Wang et al., 2023), hypernyms (Bai et al., 2022), dependency tags (Cui et al., 2022) and POS tags (Martinez et al., 2023) to implicitly encode syntactic information in word representations.

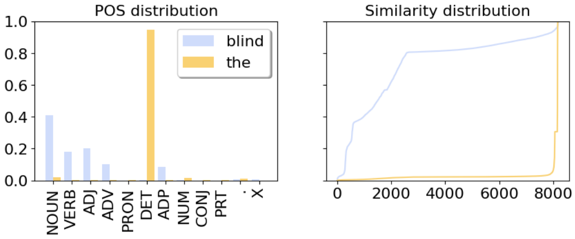

Figure 2: Part-of-speech distributions and similarities distributions for the subword tokens "blind" and "the". Similarities are computed against every other word in the vocabulary and sorted.

Here, we propose a new method for improving the representation of rare words by integrating linguistic information. This method aims to increase generalization without needing to scale up the model or data.

3 POS Smoothing

The guiding hypothesis for our method is that words that serve similar syntactic roles should receive similar gradient update steps. When any particular token is updated during training, part of the signal is distributed to all syntactically similar words using a syntactic similarity metric (operationalized below). This results in the representation of rare words approaching the average representation of all words that serve a similar syntactic function (see appendix D for an analytic explanation).

Our method consists of two components; (1) a similarity metric that uses part-of-speech distributions as a coarse proxy for syntactic similarity, and (2) an adjustment to the loss function that uses this similarity metric to smooth backpropagation over similar tokens during pre-training.

3.1 Syntactic Similarity Score

The syntactic similarity between two tokens can be measured in multiple ways, e.g., by using surface features, dependency labels, or even the predictions of a parent language model (Hinton et al., 2015). Here, we present a simple measure that considers two tokens to be similar if they have a similar distribution of part-of-speech tags in the training set.

We use the part-of-speech tagger from the NLTK package (Bird et al., 2009) to assign each word to one of the 12 universal POS tags (Petrov et al., 2012). As words can take on a different part of speech depending on the context, we count the number of times each subword token in our vocabulary V appears as each part-of-speech tag in the training data, producing a 12-valued vector. Subword tokens are assigned the part-of-speech tag of their parent word in each occurrence. This results in a matrix $M \in \mathbb{R}^{|V| \times 12}$ containing the distribution over part-of-speech tags for each token. Finally, we can compute the similarity of two tokens V_i and V_j using the cosine similarity of their part-of-speech distributions:

$$\text{POS Similarity(i, j)} = \frac{M_i^T M_j}{||M_i|| \cdot ||M_j||}.$$

This similarity metric acts as a coarse approximation for syntactic similarity. We provide the part-of-speech distributions and similarity distributions for the example tokens “blind” and “the” in fig. 2. Notice that “the” occurs almost entirely as a determiner and is not similar to most other words, whereas “blind” occurs as nouns, verbs, adjectives, and adverbs and has a high similarity to more than half of the items in the vocabulary.

3.2 Smoothing the Backpropagation Signal

Modern pre-training objectives involve likelihood maximization using cross-entropy loss between the label of the correct word and predicted probabilities from a forward pass of the model. POS Smoothing makes a small adjustment. Instead of a one-hot encoding, the target vector t becomes a distribution across the entire vocabulary with some of the signal on the correct label j and the rest of the signal distributed across all other tokens i according to the syntactic similarity metric used:

$$t_i = \begin{cases} \alpha, & \text{if } i = j \\ \frac{s(i,j)}{\sum_{k=0}^{|V|} s(i,k)} \times (1 - \alpha) & \text{otherwise} \end{cases}$$

where α determines the proportion of the error signal reserved for the correct word and s is our part-of-speech similarity metric. To emphasize the influence of syntactically similar tokens we apply a temperature smoothing function to the similarity distribution ($\tau = 0.025$). We experiment with different values for α, noting that $\alpha = 1$ is the standard likelihood maximization task. We also use a linear pacing function that gradually increases α so that at the start of training the majority of the signal is propagated to other syntactically similar tokens and by the end of training nearly all of the error signal is sent to the correct token to ensure that the model still optimizes perplexity.

We provide analytical proof (appendix D) that this minor change to the objective function results in less degenerate representations for rare words and induces a syntactic bias in all embeddings.

3.3 Experimental Setup

Generalization to rare words is particularly challenging for smaller language models and datasets (Warstadt et al., 2023; Martinez et al., 2023).

Data We use the dataset published as training data for the BabyLM challenge at the 2023 CoNLL workshop (Warstadt et al., 2023). It contains a diverse collection of transcribed speech and dialogue data, books, movie subtitles, and Wikipedia articles of roughly 10 million tokens.

Model We use a small 8-layer encoder-style RoBERTa model with pre-layer normalization (Huebner et al., 2021). We report the hyperparameter settings we use throughout our experimentation in Table 2. We use a BPE tokenizer (Sennrich et al., 2016) with a vocabulary size of 8192 as recommended in previous work (Martinez et al., 2023). We also report results for OPT-125M (Zhang et al., 2022), RoBERTa-base (Liu et al., 2019), and T5-base (Raffel et al., 2020) trained on the same dataset by Warstadt et al. (2023).

4 Results

We introduce a measure of frequency bias and show that typical language models do exhibit bias towards frequent tokens. We then show that our method reduces this bias while retaining strong language modeling capabilities.

4.1 Quantifying the Frequency Bias

We investigate frequency effects using a zero-shot test of grammatical capability known as BLiMP: The Benchmark of Linguistic Minimal Pairs (Warstadt et al., 2020). BLiMP consists of 67 datasets (or “subtasks”), each consisting of 1,000 pairs of grammatical and ungrammatical sentences that differ only with respect to a specific linguistic characteristic (covering syntax, morphology, and semantics). Language models are tasked with assigning a higher likelihood to the grammatical sentence. The grammatical generalization capabilities

Model	Bias	BLiMP	GLUE
OPT	10.6	63.2	62.7
RoBERTa	13.7	69.8	71.7
T5	6.2	58.3	58.7
Our models with POS Smoothing (PS)			
Base Model	9.8	71.4	**69.3**
PS (0.2-1.0)	5.2	72.2	68.4
PS (0.5-1.0)	5.7	72.3	68.7
PS (0.8-1.0)	7.4	71.9	**69.3**
PS (0.5-0.5)	**-0.2**	72.1	68.0
PS (0.8-0.8)	2.9	**73.2**	68.9

Table 1: We report bias (↓), BLiMP (↑), and GLUE (↑) scores for the three pretrained models, our MLM baseline, and five POS Smoothing (PS) variants. The values in parentheses indicate the α values for the pacing function at the start and end of training.

of a language model are often summarized by averaging the accuracies achieved across the 67 BLiMP tasks. With random guessing scoring 0.5, state-of-the-art models have achieved scores of 0.87 when trained on large datasets, and models trained on the 10M-word BabyLM dataset have achieved scores up to 0.8 (Warstadt et al., 2023).

BLiMP is carefully balanced to ensure individual tokens occur equally in both sentence types. However, within a single pair, there may be an imbalance in average token frequency. We hypothesize that despite the minimal difference in BLiMP pairs, models trained in a typical manner will be biased by token frequency when determining grammatical acceptability.

In each pair, we calculate the average frequency of the differing tokens over the training data. We then rank sentences according to the relative difference in average frequency (positive differences indicate higher average frequency for the acceptable sentence) and compute the BLiMP score for both the lower third and upper third of sentences. We call the difference between these two BLiMP scores the **BLiMP frequency bias** of the model tested. We illustrate this process in fig. 1.

POS Smoothing reduces frequency bias. We find that all four pretrained models exhibit strong frequency bias (see Table 1), are more likely to incorrectly prefer ungrammatical sentences if they contain tokens that occur more frequently during training. This confirms our hypothesis that the evaluation of generalization capabilities is obfuscated by frequency effects.

In contrast, all five POS Smoothing variants successfully reduce the frequency bias. The two best variants maintain a constant signal distribution throughout training (no pacing). In the case of the equal 0.5 split, the frequency bias is almost completely removed.

4.2 Language Modeling Performance

We extend our analysis beyond the specific phenomenon of the frequency bias and examine the impact of POS Smoothing on the linguistic generalization capabilities of the model and its downstream performance after finetuning.

Linguistic Generalization on BLiMP As our method aims at improving the representation of rare subwords, we did not expect a large increase in standard measures of evaluation because only relatively few test instances would be affected. In practice, however, we found that all of the POS Smoothing models achieved better BLIMP scores than our baseline model, see Table 1. These results indicate that POS Smoothing might improve the representation of all tokens, not just the rare ones.

Downstream Finetuning Effects We had concerns that softening the frequency bias with our method might lead to degraded performance in downstream tasks for which frequency can be a strong proxy. As a control condition, we finetune our model on the GLUE (Wang et al., 2018) benchmark. We find that none of the POS Smoothing objectives result in substantial performance degradation on GLUE (see the last column of Table 1).

5 Conclusion

Our work studies the phenomenon of **frequency bias** in language models that degrades the performance of these models on rare tokens. We develop a novel method for quantifying the degree to which a language model prefers grammatically incorrect sentences that contain frequent tokens over grammatically correct sentences containing infrequent tokens. We introduce a new training approach, POS Smoothing, that distributes the backpropagation signal to syntactically similar tokens. Using a coarse approximation of syntactic similarity based on part-of-speech tags, we show that this approach can remove the frequency bias without degrading downstream finetuning performance.

6 Ethical Impact

Studying long-tail data comes with some known ethical concerns. Previous research has found that names of female and non-white persons tend to fall in the long tail of many datasets which can result in models exhibiting implicit racial bias (Wolfe and Caliskan, 2021). Our paper does not directly study whether the methods we develop affect these implicit biases, although we would suspect that our approach might help remove some of these biases (without further experimentation this, however, remains a risk of our work).

Along similar lines, we also do not conduct a thorough analysis to determine whether the curated BabyLM training set we use contains offensive data or uniquely identifies individuals. For an overview of the pre-processing steps that were done to remove harmful data from the BabyLM corpora, we link the BabyLM proceedings (Warstadt et al., 2023).

We also note that the use of large-scale black-box LLMs makes studying rare word representations and their downstream effects more difficult. Our use of smaller LMs helps increase transparency and facilitates the reproducibility of our method by research groups with small computational budgets.

7 Limitations

Our methods use English-only data, and thus assume an English-centric notion of word functions. For the syntactic information, we use the POS tags provided by the NLTK tagger. As this tagger was trained on a separate dataset, this may suggest our method relies on additional data in order to best represent rare words. However, in initial experiments with an unsupervised tagger trained only on the 10M-word dataset, we achieved similar results. Finally, the models we experiment with are all relatively small and, while we assume that our results can be scaled up to larger architectures, our limited computational resources do not allow us to collect empirical evidence. In future work, we plan to further explore the impact of POS Smoothing on models with autoregressive architectures and larger training datasets.

References

He Bai, Tong Wang, Alessandro Sordoni, and Peng Shi. 2022. Better language model with hypernym class prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Dublin, Ireland. Association for Computational Linguistics.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. 2016. Morphological priors for probabilistic neural word embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 490–500, Austin, Texas. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jan Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning*, pages 1899–1907. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, Denver, Colorado. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Shijin Wang, and Ting Liu. 2022. Lert: A linguistically-motivated pre-trained language model. *arXiv preprint arXiv:2211.05344*.

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.

Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. *Advances in neural information processing systems*, 31.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *International Conference on Learning Representations*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Transformer language models handle word frequency in prediction head. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4523–4535, Toronto, Canada. Association for Computational Linguistics.

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

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 84–99, Singapore. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163.

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

Alexandre Salle and Aline Villavicencio. 2018. Incorporating subword information into matrix factorization word embeddings. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 66–71, New Orleans. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2019. Attentive mimicking: Better word embeddings by attending to informative contexts. *arXiv preprint arXiv:1904.01617*.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. 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*.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 56–68, Vancouver, Canada. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the*

2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yile Wang, Yue Zhang, Peng Li, and Yang Liu. 2023. Language model pre-training with linguistically motivated curriculum learning.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Robert Wolfe and Aylin Caliskan. 2021. Low frequency names exhibit bias and overfitting in contextualizing language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 518–532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiaosen Zheng and Jing Jiang. 2022. An empirical study of memorization in nlp. In *Annual Meeting of the Association for Computational Linguistics*.

George K. Zipf. 1935. *The Psychobiology of Language*. Boston: Houghton-Mifflin.

A Experimental Hyperparameters

These hyperparameters are taken from Martinez et al. (2023) who tuned the RoBERTa model for the 10M-word BabyLM dataset.

B Computational Requirements

We purposefully train a small-scale LM for our experiments. The total amount of the trainable parameters in our model is **12,750,336**. Each of our experiments trains for approximately 14-20 GPU hours, using a single A-100 (80GB) GPU.

Parameter	Value
Layer Norm EPS	1e-5
Learning Rate	0.001
Optimizer	AdamW
Scheduler Type	Linear
Max Steps	200,000
Warm-up Steps	50,000
Total Batch Size	512
Vocab Size	8192
Hidden Dimension Size	256
Max. Sequence Length	128
Num. Attention Layers	8
Num. Attention Heads	8
Model Architecture	RoBERTa (Pre-LN)

Table 2: Hyperparameter settings which are constant across all experiments

C Word Class Versus Word Frequency Analysis

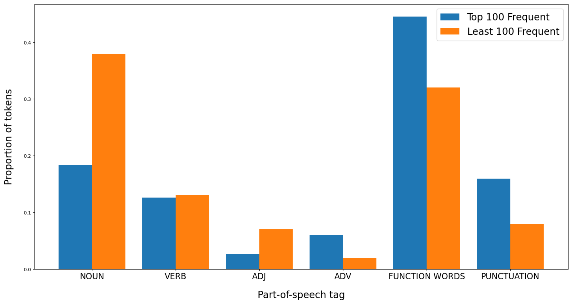

Figure 3: Distribution across POS tags of the top versus bottom 100 most frequent tokens.

Broadly, we find that content words, primarily nouns, are over-represented in low-frequency tokens. We moreover, find that the syntactic distribution across POS tags changes considerably when comparing the top 100 and bottom 100 most and least frequently occurring tokens. This analysis suggests that poor performance on rare words has a particularly strong effect on a model's inability to correctly model specialized noun vocabulary items.

D POS Smoothing Gradient Signal

To understand why the POS Smoothing technique is successful at removing frequency bias it helps to analyze how the POS Smoothing objective changes the back-propagation gradient update steps of word embeddings. One of the decisions that must be made in language modeling is whether to tie the input word embedding and output projection matrices. In recent years, common practice has become to tie the two because of both practical memory savings and theoretical findings that doing so provides a regularization method on word embeddings (Inan et al., 2017; Press and Wolf, 2017). The choice of tied versus untied output matrices, however, has a considerable effect on the gradient update steps for the word embeddings (Press and Wolf, 2017).

Let us analyze the gradient update step for the tied word embedding setting. The standard masked modeling loss function for a token at position p in an input sequence of length S is defined as

$$\mathcal{L}_p = -y_p \log(\hat{y_p}),$$

where y_p represents the true output distribution for the masked token, and $\hat{y_p} = p_\theta(y|x_{i=1:S,i\neq p})$ represents the model's predicted probability.

Let us next define some terms. We define the embedding matrix as a matrix $W \in \mathbb{R}^{|V|,S}$, where $|V|$ represents the size of the vocabulary and S represents the maximum sequence length. We also define the output of the N^{th} final attention block of a transformer as $H_N \in \mathbb{R}^{S,h_{emb}}$, with h_{emb} as the hidden embedding dimension. Let h_p represent the p^{th} row of H_N, and W_i the i^{th} row of W. Classically, transformers compute

$$\hat{y_{p_i}} = \sigma(h_p^T W_i) = \frac{e^{h_p^T W_i}}{\sum_{j\in|V|} e^{h_p^T W_j}}.$$

We then find that $\frac{\partial \mathcal{L}_p}{\partial W_i}$ can be expressed as the sum of two terms:

$$\frac{\partial \mathcal{L}_p}{\partial W_i} = \big(\sigma(h_p^T W_i) - y_p\big)h_p + \frac{\partial h_p}{\partial W_i} \qquad (1)$$

Notice that if the embedding and projection matrices are not tied, the first term of the previous equation is replaced with:

$$\frac{\partial \mathcal{L}_p}{\partial W_i} = \frac{\partial \mathcal{L}_p}{\partial h_p} + \frac{\partial h_p}{\partial W_i} \qquad (2)$$

Typically y_p, the true distribution over the correct tokens, is modeled as a point distribution (with all probability mass placed on the single masked token). Notice that equation 1 implies that when using tied-weights, the representation for token i, W_i, is updated either towards (when $h_p^T W_i < y_p$) or away (when $h_p^T W_i > y_p$) from h_p.

Now consider what occurs for a token that is rarely observed during training and when using masked language modeling. By definition, $h_p^T W_i > y_p$ will nearly always be the case and thus at every update step the representation for that token will be moved in a direction away from h_p. At the limit, this leads to rare words being pushed far away in semantic space from all other frequently occurring tokens. This exact phenomenon has been described by Gao et al. as representation degeneration.

POS Smoothing can be thought of as a method to ensure that rare words are pushed away less from and towards other tokens that serve similar syntactic functions, in the process solving the aforementioned representation degeneration problem and instead inducing a syntactic bias in the token representations.

E BLiMP Data Filtering

We filter the BLiMP data to only focus on pairs of sentences where one set of tokens has been replaced by another set and ignore sentence pairs that only differ in the order of tokens. We also remove pairs where tokens have only been added to one sentence, rather than replaced. This filtering only removes 15% of BLiMP pairs and 9 of the 67 subtasks from consideration.