Mitigating Frequency Bias and Anisotropy in Language Model Pre-Training with Syntactic Smoothing

Richard Diehl Martinez 🍊   **Zébulon Goriely** 🍊
Andrew Caines 🍊🍋   **Paula Buttery** 🍊🍋   **Lisa Beinborn** 🍏

🍊 Department of Computer Science & Technology, University of Cambridge, U.K.
🍋 ALTA Institute, University of Cambridge, U.K.
🍏 University of Göttingen, Germany

🍊 firstname.secondname@cl.cam.ac.uk   🍏 lisa.beinborn@uni-goettingen.de

Abstract

Language models strongly rely on frequency information because they maximize the likelihood of tokens during pre-training. As a consequence, language models tend not to generalize well to tokens that are seldom seen during training. Moreover, maximum likelihood training has been discovered to give rise to anisotropy: representations of tokens in a model tend to cluster tightly in a high-dimensional cone, rather than spreading out over their representational capacity.

Our work introduces a method for quantifying the **frequency bias** of a language model by assessing sentence-level perplexity with respect to token-level frequency. We then present a method for reducing the frequency bias of a language model by inducing a syntactic prior over token representations during pre-training. Our `Syntactic Smoothing` method adjusts the maximum likelihood objective function to distribute the learning signal to syntactically similar tokens. This approach results in better performance on infrequent English tokens and a decrease in anisotropy. We empirically show that the degree of anisotropy in a model correlates with its frequency bias.

⭘ | rdiehlmartinez/syntactic-smoothing

1 Introduction

Humans possess a remarkable ability to quickly understand the meaning of unknown words, given contextual cues. Consider the sentence, "the Golden Gate Bridge has been *obnebulated* every morning this week, limiting visibility of the Pacific Ocean." For many readers, 'obnebulated' is probably not a familiar term, but we are likely to infer that 1) it is almost certainly a verb because of the -ed suffix and occurrence after perfective and passive auxiliaries, and 2) its meaning relates to visibility and climatic conditions.[1] The

---

[1] 'Obnebulate' is an obsolete word meaning, "To obscure with or as with a mist; to befog" (*Oxford English Dictionary*).
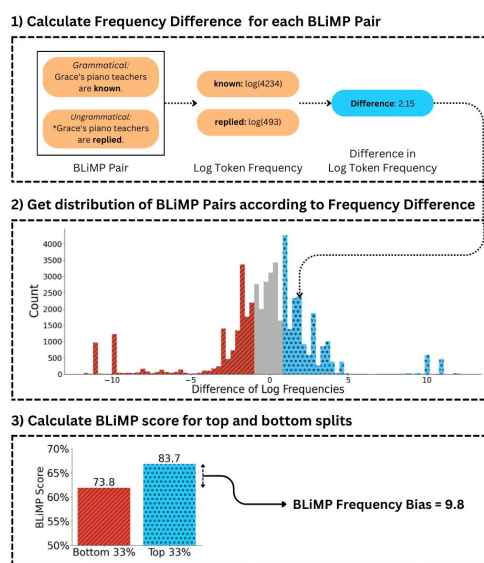

Figure 1: Illustration of the BLiMP **frequency bias** calculation used to evaluate a model's reliance on frequency statistics when making predictions. The example BLiMP values are from a baseline RoBERTa model.

ability to integrate unknown words based on syntactic and semantic context is essential for robust language understanding and still poses a significant challenge for language models. Nevertheless, Pre-trained Transformer Language Models (PLMs) have proven tremendously capable of solving a wide array of language processing tasks (Touvron et al., 2023; Chowdhery et al., 2023).

Part of the success of PLMs can be attributed to the pre-training objective. Despite variations 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). As language use is characterized by a Zipfian distribution (Zipf, 1935), language models are exposed to frequent tokens exponentially more often than infrequent ones during pre-training. Consequently, the representations of these frequent tokens are optimized based on

exponentially more learning signals than those of low-frequency tokens. It has been shown that maximum likelihood objectives lead to representation degeneration in English language models because infrequent tokens are pushed into a narrow manifold of the representational space (Gao et al., 2019). This representation degeneration problem is linked to the broader problem of **anisotropy**: the hidden states of a language model tend to cluster together into a small cone-shaped subspace, rather than over their full representational capacity (Arora et al., 2016a; Ethayarajh, 2019; Gao et al., 2019). As language model evaluation is based on cumulative evaluation scores that conceal how well a model processes infrequent words, the disparities in the representational space are difficult to assess.

Conventional language modeling approaches require large model sizes to effectively capture long-tail vocabulary distributions, limiting the scalability of these methods (Feldman, 2020; Haviv et al., 2023). In this work, we propose `Syntactic Smoothing`: a syntactically-guided label smoothing approach to improve the representation of infrequent tokens in language models without resorting to perpetual increases of model and training data size. We smoothly distribute the backpropagation signal over syntactically similar tokens using a similarity metric based on part-of-speech (POS) tag distributions. Using this method, tokens that are seldom seen during training benefit from the more frequent updates of tokens that occur in similar syntactic functions. We evaluate our method using a new metric for quantifying the **frequency bias** of language models (illustrated in fig. 1) and find that `Syntactic Smoothing` reduces both the frequency bias and the degree of anisotropy in a small English language model. We further explore the relationship between anisotropy and frequency bias and their effect on downstream performance.

2 Related Literature

Through maximum likelihood training, language models implicitly learn to encode token frequency statistics. This training process gives rise to a frequency bias in models that constrains their ability to generalize to infrequent tokens. In this section, we begin by reviewing literature that discusses the challenges of generalizing linguistic knowledge to infrequent tokens. We then examine recent work that links the impact of token frequency to anisotropy in the models' representational space.

2.1 Generalization to Infrequent Tokens

Current approaches to language modeling rely heavily on the memorization of infrequent tokens to perform well on downstream tasks (Feldman, 2020). Recent analytical work has shown that certain layers of transformer models implicitly store memorized long-tail data (Haviv et al., 2023; Kobayashi 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 (Belkin et al., 2019). While these studies present various metrics to evaluate memorization, these metrics do not capture how memorization impacts generalized linguistic understanding within the models. In our work, we address this gap by developing a metric that quantifies the extent of this frequency bias in relation to models' linguistic abilities.

Language use follows a Zipfian distribution, meaning that many tokens appear infrequently. Standard training objectives often require large models and noisy datasets with sufficient long-tail samples for effective generalization (Zheng and Jiang, 2022). However, improving generalization without excessive scaling can be achieved by training models with inductive priors that leverage linguistic information. On the lexical level, the integration of morphological and orthographic information during representation learning has been explored to obtain more fine-grained word embeddings (Salle and Villavicencio, 2018; Vulić et al., 2017; Cotterell and Schütze, 2015; Bhatia et al., 2016; Botha and Blunsom, 2014). To improve syntactic generalization, the objective function has been enriched with auxiliary tasks, such as predicting constituency labels (Wang et al., 2023), hypernyms (Bai et al., 2022), dependency tags (Cui et al., 2022), and POS tags (Diehl Martinez et al., 2023). Some approaches have also shown promising results on rare word performance by constructing token embeddings that consider a word's surface form and surrounding context (Schick and Schütze, 2019, 2020).

2.2 Anisotropy in Representational Space

While frequency bias and generalization capabilities can be observed by analyzing model behavior on input–output patterns, representational analyses indicate that these phenomena are linked to the

distribution of token representations. Language models trained as likelihood maximizers have been shown to yield degenerate representations for rare tokens (Gao et al., 2019). Throughout training, infrequent tokens are disproportionately pushed in the negative direction of most hidden states, resulting in their clustering together irrespective of their semantic or syntactic properties. This clustering behavior leads to anisotropy: rather than occupying a large region of the representational space, token representations lie along a narrow manifold (Gao et al., 2019; Ethayarajh, 2019).

2.2.1 Defining Anisotropy

Anisotropy is defined as the inverse of isotropy: $1 - I(v(\cdot))$. A representational space is isotropic if all the vector directions are distributed uniformly, meaning no particular direction is favored over another.

Arora et al. and Mu and Viswanath define isotropy as:

$$I(v(\cdot)) := \frac{\min_{\|c\|=1} Z(c)}{\max_{\|c\|=1} Z(c)} \quad (1)$$

where c is a unit vector and $Z(c)$ is defined as the partition function over all tokens w in the vocabulary V, with representations $v(w)$:

$$Z(c) = \sum_{w \in V} \exp(c^T v(w))$$

In practice, this definition of isotropy is analytically infeasible to solve. In this paper, we follow an empirical approximation proposed by Ethayarajh:

$$I(v(\cdot)) := \mathbb{E}_{i \neq j}\big(1 - \cos(v(w_i), v(w_j))\big) \quad (2)$$

Here, w_i and w_j are two tokens sampled from the vocabulary, and $\cos$ is defined as taking the cosine similarity of the two word representations for w_i and w_j.

Despite its prevalence, the impact of anisotropy on a model's language understanding abilities remains unclear. Some studies suggest that reducing anisotropy improves performance on noncontextual benchmarks, sentence comparison tasks, and multilingual benchmarks (Biś et al., 2021; Su et al., 2021; Rajaee and Pilehvar, 2022). Conversely, other research indicates that higher anisotropy might enhance semantic clustering tasks and that reducing anisotropy does not uniformly improve performance on common NLU tasks (Ait-Saada and Nadif, 2023; Ding et al., 2022). Furthermore, the relationship between anisotropy and

maximum likelihood training has been questioned. Some researchers argue that isotropy exists in local manifolds of contextual word representations (Cai et al., 2020), while others contend that anisotropy arises from the learning dynamics of the query and key attention matrices in transformer models (Godey et al., 2024).

2.2.2 Reducing Anisotropy

Existing methods to reduce anisotropy broadly fall into three categories. The first group of approaches transforms the hidden states of language models to remove semantically uninformative directions and to preserve the dimensions of maximal isotropy (Arora et al., 2016b; Mu and Viswanath, 2018; Raunak et al., 2019; Su et al., 2021; Biś et al., 2021). This intervention style is based on the assumption that the top singular dimensions of pre-trained word representations encode frequency statistics rather than semantic or lexical information (Mu and Viswanath, 2018). The second category of methods introduces novel training objectives and regularization terms that reduce the effects of anisotropy (Gong et al., 2018; Gao et al., 2019; Wang et al., 2019). This type of approach places an inductive bias on representations that push the embeddings of frequent and infrequent words to occupy a similar semantic space. The third set of approaches explores different training paradigms to directly minimize anisotropy, such as using normalizing flow models (Li et al., 2020) or manipulating the gradients used in maximum likelihood models (Yu et al., 2022)

While frequency bias and anisotropy are prevalent in language modeling, quantifying their effects and understanding their impact on generalization, particularly for infrequent words, remains an open area of research. Our paper introduces a novel method for improving the representation of infrequent tokens by integrating linguistic information. Moreover, we hypothesize that adjusting the learning process to better represent infrequent tokens will also reduce anisotropy, as these two phenomena are interconnected.

3 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 comprises 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 of a language model are often summarized by averaging the accuracies achieved across the 67 BLiMP tasks. While random guessing scores 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 0.80 (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: For instance, the sentence *Grace's piano teachers are **known*** has a log frequency of 8.35 while its associated minimal pair *Grace's piano teachers are **replied*** has a log frequency of 6.20. 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.

Our goal is to quantify how language model performance differs between BLIMP pairs with large positive frequency differences (where the correct sentence has more frequently occurring tokens) and with large negative frequency differences (where the correct sentence has much less frequently occurring tokens). We do so in two steps.

First, for each BLIMP sentence pair, we calculate the average (natural log) frequency of the differing tokens. Frequencies of individual tokens are computed with respect to a model's training data; for instance, in the example above the token **known** has a log frequency of 8.35 in the training data. Sentence pairs are then ranked by the relative difference in these average frequencies, where positive values indicate a higher average frequency for the acceptable sentence. These relative differences form a distribution, as shown in the middle plot of fig. 1.

Then, we compute the BLiMP score using pseudo log-likelihood (Salazar et al., 2020) for BLIMP pairs in the upper and lower thirds of the relative frequency difference distribution. We exclude the middle third, as these represent pairs with minimal frequency differences (see the frequency plot for details). We define a model's **frequency**

bias as the difference between the two BLiMP scores. The entire process is illustrated in fig. 1.

In practice, we find that standard transformer language models, such as OPT-125M (Zhang et al., 2022), RoBERTa-base (Liu et al., 2019), and T5-base (Raffel et al., 2020), exhibit a frequency bias as high as 13.7%. Our goal is to develop a model that can attain a frequency bias close to zero while attaining a high BLiMP score: that is, a model that makes determinations on the grammatical acceptability of sentences based solely on relevant linguistic aspects, rather than relying on possibly misleading statistical artifacts of the training data.

4 Syntactic Smoothing

We hypothesize that transformer language models exhibit a strong frequency bias due to their maximum likelihood training objective, which limits infrequent tokens from receiving useful learning signals and thus hinders their ability to effectively encode linguistic information. To address this, we propose at each learning step to backpropagate the learning signal of a target token to all other tokens serving similar syntactic roles; this benefits infrequent tokens that appear less often in the training data.

`Syntactic Smoothing` implements this strategy by distributing a portion of every update signal to all syntactically similar tokens using a syntactic similarity metric (operationalized below). This results in the representation of infrequent tokens approaching the average representation of all tokens that serve a similar syntactic function; e.g., the representation of a niche word like 'obnebulated' would encode its syntactic role as a verb.

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 to smooth the backpropagation signal over syntactically similar tokens during pre-training.

4.1 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 teacher language model (Hinton et al., 2015). Here, we present a simple measure that acts as a coarse approximation for syntactic similarity: we consider two tokens to be similar if they have a similar distribution of part-of-speech tags in the

training set.

We evaluate the syntactic similarity between tokens prior to training, as a one-off preprocessing step over the entire training set. First, we use the part-of-speech (POS) tagger from the NLTK package (Bird et al., 2009) to assign each word in the training set to one of 12 universal POS tags, based on its given context (Petrov et al., 2012).[2] We then tokenize the training data into sub-word tokens and assign each token the POS tag corresponding to the word it belongs to in each instance. As words can take on a different part of speech depending on the context, we count the number of times each token in our vocabulary V appears as each POS tag in the training data, producing a 12-valued vector. This results in a matrix $M \in \mathbb{R}^{|V| \times 12}$ containing the distribution over POS tags for each token. Finally, we can compute the similarity of two tokens V_i and V_j using the cosine similarity of their POS distributions:

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

Note that while in this paper we define syntactic similarity via cosine similarity, any real-valued distance metric or divergence can be used. The similarity function does not need to be symmetric, although we note that symmetric functions provide computational advantages as only half the values need to be computed and stored. Also, note that our methodology does not depend on a specific choice of POS tagger.

We provide the POS distributions and similarity distributions for the example tokens "blind" and "the" in fig. 2. Notice that "the" occurs almost exclusively as a determiner and is not similar to many other tokens, whereas "blind" occurs as a noun, verb, adjective, and adverb and has a high similarity to more than half the other tokens in the vocabulary.

4.2 Smoothing the Backpropagation Signal

Modern pre-training objectives implement likelihood maximization using a cross-entropy loss between the label of the correct word and predicted probabilities from a forward pass of the model. `Syntactic Smoothing` makes a small adjustment.
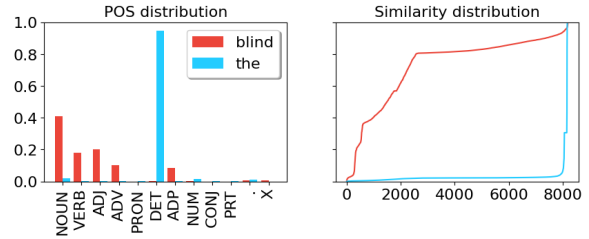
Figure 2: Part-of-speech distributions and similarity distributions for the subword tokens "blind" and "the". Similarities are computed as cosine-similarities against every other token in the vocabulary and sorted.

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:

$$t_i = \begin{cases} (1 - \alpha), & \text{if } i = j \\ \frac{s(i,j)}{\sum_{k=0}^{|V|} s(i,k)} \times \alpha & \text{otherwise} \end{cases} \quad (3)$$

where α, the smoothing parameter, determines the proportion of the error signal reserved for the correct word and s is our part-of-speech similarity metric. We experiment with different values for α, noting that $\alpha = 0$ is the standard likelihood maximization task. We also investigate the use of a pacing function that linearly decreases 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.

In practice, we also find it beneficial to apply a temperature scaling function to the syntactic similarity distribution. Thus, rather than using the raw syntactic similarity scores, $s(i, j)$, in eq. (3), we use the temperature-scaled similarity scores:

$$s'(i, j) = \frac{\exp\left(\frac{s(i,j)}{\tau}\right)}{\sum_{k=1}^{|V|} \exp\left(\frac{s(i,k)}{\tau}\right)}$$

where τ defines the temperature which we set to $\tau = 0.025$.

4.3 Experimental Setup

Our experiments focus on smaller language models and datasets due to computational constraints and the particular challenges of generalizing to uncommon instances under resource-constrained training conditions (Warstadt et al., 2023; Diehl 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 roughly 10 million tokens sampled from pre-existing datasets, covering a wide range of domains including transcribed speech (both adult-directed and child-directed), movie subtitles, Wikipedia articles, and books. The dataset was constructed to be similar to the input received by children — 56% comes from transcribed speech and 40% comes from sources intended for children.

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 all experiments in table 3 (appendix A) and computational requirements in appendix B. We use a BPE tokenizer (Sennrich et al., 2016) with a vocabulary size of 8192 as recommended in previous work (Diehl Martinez et al., 2023).

Evaluation We evaluate the BLiMP frequency bias of our models, as defined in section 3, on the evaluation set of BLiMP. To compute anisotropy we use the formulation defined in eq. (2); We sample 1,000 pairs of random word tokens with their surrounding context from the training set, and compute the cosine similarity of their hidden representation at each of the 8 layers of the RoBERTa model. To obtain a model's final anisotropy value, we average the anisotropy scores across the 8 layers. Additionally, we finetune and evaluate each model on two downstream sentence-level tasks, COLA (Warstadt et al., 2019) and SST-2 (Socher et al., 2013), as well as two language inference tasks, MNLI (Williams et al., 2018) and QNLI (Rajpurkar et al., 2016; Wang et al., 2018).

Baselines We introduce three types of baselines:

1. **Popular open-source transformer models**: OPT-125M (Zhang et al., 2022), RoBERTa-base (Liu et al., 2019), and T5-base (Raffel et al., 2020), pre-trained from scratch on the same dataset we describe in section 4.3. We use the default configuration for each model resulting in a varied number of parameters.

2. **Base Model**: The small RoBERTa model described above without Syntactic Smoothing.

3. **Label Smoothing**: The base model trained with label smoothing (Szegedy et al., 2016).

We train a baseline with a low-level of smoothing ($\alpha = 0.2$) and a mid-level of smoothing ($\alpha = 0.5$). Note that Syntactic Smoothing can be seen as a linguistically-guided version of the standard label smoothing approach, in which the learning signal is distributed to all tokens uniformly.

Our Models We train our models with Syntactic Smoothing using the same two α values as the label smoothing baselines to facilitate comparison. We also run variants using the linear pacing function presented in section 4.2 which linearly decreases the smoothing from an initial value of α to zero across training. For these variants, we use the same two values of smoothing, as well as an additional high value of $\alpha = 0.8$ giving a total of five Syntactic Smoothing variants.[3]

5 Results

Our results are summarized in table 1. We find that our method reduces frequency bias while retaining strong language modeling capabilities. At the same time, we observe that the models with the lowest frequency bias also demonstrate the lowest anisotropy. We then extend our analysis beyond the specific phenomenon of frequency bias and anisotropy by examining the impact of Syntactic Smoothing on the linguistic generalization capabilities of the model and its downstream performance after finetuning. Finally, we find that an alternative syntactic scoring metric leads to similar results as the cosine-based definition.

5.1 Anisotropy and Frequency Bias

We conduct analyses to inspect the learning dynamics of our method and its effect on frequency bias and anisotropy in more detail.

Syntactic Smoothing reduces frequency bias. We find that all four pre-trained models exhibit strong frequency bias (see fig. 3); they 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.

---

[3]We do not include unpaced Syntactic Smoothing with a high value of α as initial experiments found that distributing such a high proportion of the learning signal away from the correct token leads to high perplexity and poor downstream performance.

Model	α	Bias	Anisotropy	BLiMP	COLA	SST-2	MNLI	QNLI
Base Model	-	9.8	51.3	71.4	71.4	82.9	69.6	79.7
Label Smoothing	Low	5.5	40.2	73.2	70.7	84.0	**70.1**	80.0
	Mid	2.7	40.3	73.0	71.5	82.2	69.0	79.4
Syntactic Smoothing	Low	2.9	39.7	**73.2**	70.7	84.9	69.7	79.2
	Mid	-0.2	33.8	72.1	**71.9**	83.5	67.2	79.4
	Paced Low	7.4	39.9	71.9	70.5	**85.2**	70.0	**80.4**
	Paced Mid	5.7	34.5	72.3	71.8	84.0	68.2	78.9
	Paced High	5.2	**31.0**	72.2	70.5	83.7	67.7	79.1

Table 1: We report bias (↓), anisotropy (↓), BLiMP (↑) score, and accuracy or correlation scores (↑) on two downstream sentence-level tasks – COLA and SST-2 – and two downstream language inference tasks – MNLI and QNLI – for our MLM baseline, two label smoothing (LS) baselines, and five Syntactic Smoothing (SyS) variants. SyS-P variants use linear pacing to reduce the smoothing factor to zero over training.
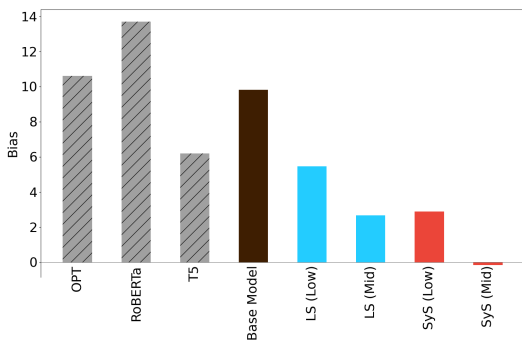

Figure 3: Frequency bias plotted for the three open source pre-trained models, our base model, the two label smoothing (LS) baselines and our two Syntactic Smoothing (SyS) models.

By contrast, the two Syntactic Smoothing variants successfully reduce the frequency bias. The frequency bias is almost completely removed in the case of the Mid variant, which distributes exactly half of the training signal to syntactically similar tokens. We further observe that the Label Smoothing baselines also reduce bias but to a lesser extent than the corresponding Syntactic Smoothing models with the same degree of smoothing.
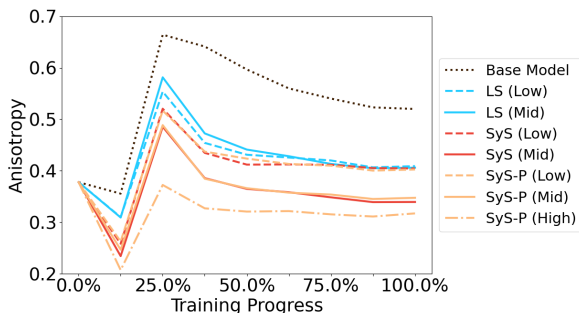

Figure 4: Anisotropy learning dynamics plotted for the baseline RoBERTa model, the two label smoothing (LS) baselines and our Syntactic Smoothing (SyS) models. Values in parentheses indicate the degree of smoothing.

Syntactic Smoothing reduces anisotropy. As shown in table 1, Syntactic Smoothing reduces anisotropy over both the base model and label smoothing baselines.[4] Label smoothing reduces anisotropy, but not to the same extent as our Syntactic Smoothing models. To better understand how anisotropy develops in a model, we compute the model's anisotropy scores at eight checkpoints during training, as shown in fig. 4. We find that a greater degree of smoothing leads to a greater reduction in anisotropy for our Syntactic Smoothing variants (it is less clear if this is the case for label smoothing), supporting our hypothesis that syntactic initialization helps promote better representation learning across the model's vocabulary. We also find that the pacing method leads to lower anisotropy than the flat method, with SyS-P (High) achieving the lowest anisotropy throughout.

Over the course of training, we observe a consistent double-dip trend: an initial dip followed by a sudden rise, followed by a second slow decrease in anisotropy. The Syntactic Smoothing models do not see as large a sudden rise, maintaining a lower anisotropy throughout. To examine the learning dynamics in more detail, we also plot the evolution of the anisotropy across several layers of our baseline model and the SyS-P (High) variant, given in fig. 5. Two observations stand out. The anisotropy of all layers in the Syntactic Smoothing model is lower than in the corresponding layers in the baseline model across the entire learning process. In both the baseline model and the Syntactic Smoothing model, earlier layers have lower anisotropy; this finding agrees with the

---

[4]Note that we do not compute the anisotropy for the three open-source pre-trained models (OPT, RoBERTa, T5) because these models use different architectural configurations than the models we train (e.g., larger hidden dimensions).

same observation made by Ethayarajh. Notably, in the final layer—commonly used for sentence representations in downstream tasks—the anisotropy of the `Syntactic Smoothing` model remains consistently low and does not increase significantly during training, in contrast to the drastic fluctuation observed in the baseline model.
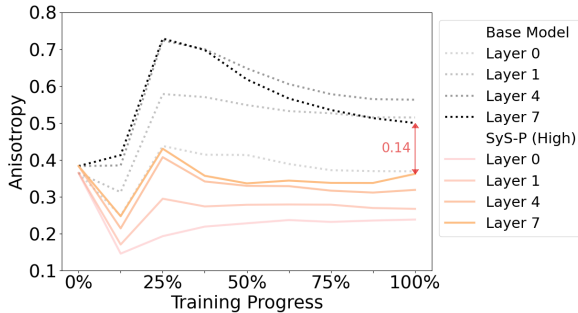

Figure 5: Anisotropy learning dynamics plotted for the baseline model and the paced `Syntactic Smoothing` model with high smoothing, across some of the models' layers. We highlight the difference in anisotropy of the final layer across the two models at the end of training.

Frequency bias and anisotropy are correlated. For each model, we compute the model's frequency bias and anisotropy at multiple training stages. We plot the learning dynamics of anisotropy and frequency bias in fig. 6, only including the points after 50% of training has been completed to avoid the noisy first dip observed in the anisotropy dynamics above. We find a positive Pearson correlation of 0.73 and a polynomial goodness-of-fit R^2 score of 0.63 between these two metrics.

It is also evident that the pacing approach reintroduces frequency bias towards the end of training, as the degree of smoothing is linearly reduced to zero. It is noteworthy that the final anisotropy and bias are lower than the baseline model, and completing training without any smoothing may be beneficial for downstream tasks, as explored in the next section.

5.2 Effects of Smoothing on Downstream Tasks

While our method primarily aims to enhance the representation of infrequent tokens, we sought to investigate the potential for improvement in standard evaluation measures, given the limited number of affected test instances. Nonetheless, we observe that all the `Syntactic Smoothing` models, as well as the label smoothing models, achieve better BLIMP scores than our baseline model (see
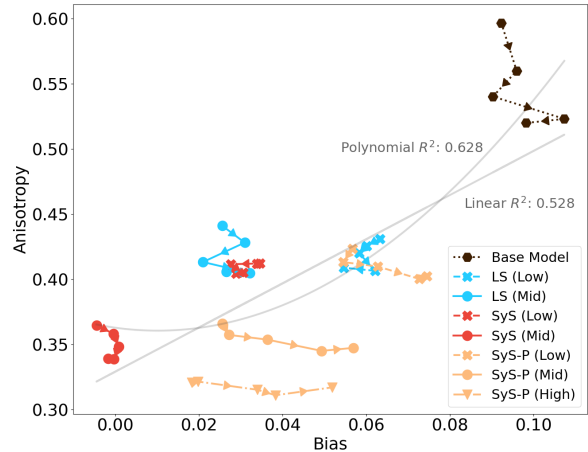

Figure 6: Pairs of anisotropy, and frequency bias for the baseline RoBERTa model, the two label smoothing baselines and our `Syntactic Smoothing` models. The arrows indicate increasing training progress (starting after 50% of training has completed).

table 1). These results suggest that methods that smooth label distributions, whether through a syntactic prior or a simpler uniform smoothing approach, enhance the representation of all tokens, including the more frequent ones.

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 two sentence-level tasks (COLA, SST-2) and two language inference tasks (MNLI and QNLI), both of which are part of the GLUE (Wang et al., 2018) benchmark. We find that none of the `Syntactic Smoothing` objectives result in substantial performance degradation on these NLU tasks (see the last four columns of Table table 1), and in fact note that for some tasks, such as SST-2, the `Syntactic Smoothing` models yield uniform increases in performance.[5]

5.3 Alternative Measures of Syntactic Similarity

In section 4.1 we define the syntactic similarity score that is used by the `Syntactic Smoothing` approach as the cosine similarity between POS distributions. To examine how this specific choice of similarity metric impacts our approach, we replace the cosine-based definition with a Jensen Shannon-based definition:

$$\frac{1}{2}\big[\text{KL}(M_i, M_j) + \text{KL}(M_j, M_i)\big],$$

---

[5]While not comparable apples-to-apples, we report NLU performance for the open-source baselines in appendix E.

6006

Model		Bias	Anisotropy	BLiMP
Base Model	‖	9.8	51.3	71.4
SyS (Mid)	[JS] ‖	3.6	34.7	71.3
SyS (Low)	[JS] ‖	4.1	34.6	73.3
SyS-P (High)	[JS] ‖	6.6	36.7	72.5
SyS-P (Mid)	[JS] ‖	8.4	39.1	73.0
SyS-P (Low)	[JS] ‖	5.0	34.5	72.9

Table 2: Results for bias (↓), anisotropy (↓), and BLiMP (↑) score for Syntactic Smoothing (SyS) models that use a Jensen Shannon-based [JS] definition of the similarity metric.

where $\text{KL}(M_i, M_j)$ is the Kullback-Leibler divergence between the POS distributions, M_i and M_j, for the vocabulary items V_i and V_j.

Summarized in table 2, we note that the effect of using a Jensen Shannon-based definition of the similarity metric yields a similar (albeit slightly smaller) decrease in frequency bias and anisotropy, as compared to the standard cosine-based definition of the similarity metric.

6 Conclusion

Our work studies the phenomenon of **frequency bias** in language models that degrades the performance of these models on tokens infrequently observed during training. 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, Syntactic 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 broader language understanding. We also find that reductions in frequency bias are strongly correlated with reductions in a model's anisotropy. Our findings provide a novel angle through which to observe the role of anisotropy in language modeling.

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 less efficient neural representations of these names compared to names of male and white persons(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 refer the reader to the BabyLM proceedings (Warstadt et al., 2023).

We also note that the use of large-scale black-box LLMs makes studying infrequent token 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.

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 infrequent words. However, in initial experiments with an unsupervised tagger trained only on the 10M-word dataset, we achieved similar results. Additionally, 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 Syntactic Smoothing on models with autoregressive architectures and larger training datasets. We also hope future work will apply our method to more languages, possibly leveraging unsupervised POS taggers for these languages, and evaluate the effect of Syntactic Smoothing on different downstream tasks (particularly tasks with irregular vocabulary frequency distributions).

Acknowledgements

References

Mira Ait-Saada and Mohamed Nadif. 2023. Is anisotropy truly harmful? a case study on text clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1194–1203, Toronto, Canada. Association for Computational Linguistics.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016a. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016b. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

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.".

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online. Association for Computational Linguistics.

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.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2020. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*.

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.

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.

Yue Ding, Karolis Martinkus, Damian Pascual, Simon Clematide, and Roger Wattenhofer. 2022. On

isotropy calibration of transformer models. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 1–9.

Kawin Ethayarajh. 2019. How contextualized 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, Hong Kong, China. 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*.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2024. Anisotropy is inherent to self-attention in transformers. *arXiv preprint arXiv:2401.12143*.

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, Dubrovnik, Croatia. Association for Computational Linguistics.

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.

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.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

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*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

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).

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.

Sara Rajaee and Mohammad Taher Pilehvar. 2022. An isotropy analysis in the multilingual BERT embedding space. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1309–1316, Dublin, Ireland. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

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, and Ryan Cotterell. 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 1–34, Singapore. Association for Computational Linguistics.

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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. 2022. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29–45, Dublin, Ireland. Association for Computational Linguistics.

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

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 3: Hyperparameter settings which are constant across all experiments

These hyperparameters are taken from Diehl 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 server with one NVIDIA A100 80GB PCIe GPU, 32 CPUs, and 32 GB of RAM for all experiments. Below, we report a subset of the output of the *lscpu* command:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical,
                       48 bits virtual
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R)
                       Silver 4210R CPU
                       @ 2.40GHz
CPU family:            6
Model:                 85
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             8
Stepping:              7
BogoMIPS:              4800.11
```

C Word Class Versus Word Frequency Analysis

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 infrequent tokens has a particularly strong effect on a model's inability to correctly model specialized noun vocabulary items.
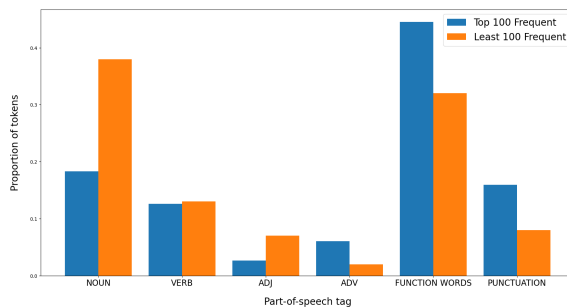

Figure 7: Distribution across POS tags of the top versus bottom 100 most frequent tokens.

D 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.

E NLU Performance of Open-Source Baselines

Model	BLiMP	COLA	SST-2	MNLI	QNLI
OPT	63.2	64.6	81.9	57.6	61.5
RoBERTa	69.8	70.8	87.0	73.2	77.0
T5	58.3	61.2	78.1	48.0	62.0

Table 4: BLiMP (↑) score and accuracy (↑) on sentence-level tasks (COLA, SST-2) and language inference tasks (MNLI, QNLI) for the three open-source transformer baselines.