Too Much in Common: Shifting of Embeddings in Transformer Language Models and its Implications

Daniel Biś
Florida State University
Tallahassee, USA
bis@cs.fsu.edu

Maksim Podkorytov
Florida State University
Tallahassee, USA
maksim@cs.fsu.edu

Xiuwen Liu
Florida State University
Tallahassee, USA
liux@cs.fsu.edu

Abstract

The success of language models based on the Transformer architecture appears to be inconsistent with observed anisotropic properties of representations learned by such models. We resolve this by showing, contrary to previous studies, that the representations do not occupy a narrow cone, but rather drift in common directions. At any training step, all of the embeddings except for the ground-truth target embedding are updated with gradient in the same direction. Compounded over the training set, the embeddings drift and share common components, manifested in their shape in all the models we have empirically tested. Our experiments show that isotropy can be restored using a simple transformation.[1]

1 Introduction

Word embeddings, both static (Mikolov et al., 2013a; Pennington et al., 2014) and contextualized (Peters et al., 2018), have been instrumental to the progress made in Natural Language Processing over the past decade (Turian et al., 2010; Wu et al., 2016; Liu et al., 2018; Peters et al., 2018; Devlin et al., 2019). In recent years, language models based on Transformer architecture (Vaswani et al., 2017) have led to state-of-the-art performance on problems such as machine translation (Vaswani et al., 2017), question answering (Devlin et al., 2019; Liu et al., 2019b), and Word Sense Disambiguation (Bevilacqua and Navigli, 2020), among others. However, it has been observed that representations from Transformers exhibit undesirable properties, such as anisotropy, that is tend to occupy only a small subspace of the embedding space. The observation has been documented by a number of studies (Gao et al., 2019; Ethayarajh, 2019; Wang et al., 2020). A similar property has been identified in the past in static word embeddings (Mu and Viswanath, 2018). To address the issues, post-processing methods (Mu and Viswanath, 2018), and regularization terms have been proposed (Gao et al., 2019; Wang et al., 2019c, 2020). However, the mechanism that leads to undesirable properties remains unclear. Without understanding the mechanism, it is going to be difficult to address the fundamental issue properly.

The deficiencies are most pronounced in the representations of rare words, as we will show in Section 4. Performance of pretrained language models is inconsistent and tends to decrease when input contains rare words (Schick and Schütze, 2020b,a). Schick and Schütze (2020a) observe that replacing a portion of words in the MNLI (Williams et al., 2018) entailment data set with less frequent synonyms leads to decrease in performance of BERT-base and RoBERTa-large by 30% and 21.8% respectively.[2] After enriching rare words with surface-form features and additional context, Schick and Schütze (2020a) decrease the performance gap to 20.7% for BERT and 17% for RoBERTa, but the gap remains large nonetheless. Why do even the large-scale, pretrained language models struggle to learn good representations of rare words? Consider a language model with an embedding matrix shared between the input and output layers, a standard setup known as weight tying trick (Inan et al., 2017). Intuitively, at any training step t, optimization of the cross-entropy loss can be characterized as "pulling" the target embedding, $\boldsymbol{w}_T$, closer to the model's output vector $\boldsymbol{h}_t$, while "pushing" all other embeddings, $\boldsymbol{W} \setminus \boldsymbol{w}_T$, in the same direction, away from the output vector $\boldsymbol{h}_t$. This leads to what we call *common enemies effect* – the effect of the target words producing gradients of the same direction for all of the non-target words. Compounded over the training set, the embeddings drift and share common components, manifested in their shape in all the models

[1] The code and datasets used in this paper are available at https://github.com/danielbis/tooMuchInCommon.

[2] Based on the results reported by authors.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5117–5130
June 6–11, 2021.

we have empirically tested; see Figure 1.

Although Gao et al. (2019) report a closely related phenomenon and call it *representation degeneration*, their analysis is based on an assumption that the embedding matrix is learned after all other parameters of the model are well-optimized and fixed, which is not the case in practice. We conduct our analysis in a more realistic setting, and arrive at different conclusions. We show that embeddings do not occupy a narrow cone, but are shifted in one common direction and only appear as a cone when projected to a lower dimensional space (Section 4.1). In fact simply removing the mean vector of all embeddings, thus centering them, shifts the embeddings back onto a more spherical shape. We evaluate embeddings, before and after centering, on four standard benchmarks and observe significant performance improvement across all of them. Why is removing the mean so effective? We find that the common enemies effect applies to most, if not all, words in the vocabulary but in non-uniform manner. As language is known to follow an approximately Zipfian distribution (Zipf, 1949; Manning and Schütze, 2001; Piantadosi, 2014) even common words will not occur frequently in a text corpus, and in result will be often "pushed" by other target words in the same direction as rare words. Consequently, all embeddings share a significant common direction. We will focus on the analysis of auto-regressive GPT-2 (Radford et al., 2019) and two masked language models, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b). Our contributions can be summarized as follows:

- We show that as word embeddings repeatedly share same direction gradients, they are shifted in one dominant direction in the vector space. The effects are the most evident in representations of rare words, but are also present in representations of frequent words.

- The shift causes the distribution of projected embeddings to appear as a narrow cone; we show that simply removing the mean vector is enough to restore the spherical distribution.

- We provide empirical evidence of our analyses using state-of-the-art pretrained language models and demonstrate that removing the mean dramatically improves isotropy of the representations.

2 Background

2.1 Distributed Word Representations

Distributed representations induce a rich similarity space, in which semantically similar concepts are close in distance (Goodfellow et al., 2016; Bengio et al., 2003; Mikolov et al., 2013c). In a language model, the regularities of embeddings space facilitate generalization, assigning a high probability to a sequence of words that has never been seen before but consists of words that are similar to words forming an already seen sentence (Bengio et al., 2003; Mikolov et al., 2013c). Although models such as BERT or GPT-2 produce representations from a function of the entire input sequence, the representations are a result of a series of transformations applied to the input vectors. Consider an example sentence: *"The building was dilapidated."*, and the sentences resulting from replacing *"dilapidated"* with either *"ruined"* or *"reconditioned"*. If the distance in the embeddings space between the two rather infrequent, but antonymous, words *"dilapidated"* and *"reconditioned"* is not larger than the distance between *"dilapidated"* and its relatively frequent synonym *"ruined"*, then by the aforementioned generalization principle there is little to no reason to believe that the distance will become larger in the output layer.[3]

2.2 Tokenization

Do the subword tokenization methods (Schuster and Nakajima, 2012; Wu et al., 2016; Sennrich et al., 2016; Radford et al., 2019) preserve the word frequency imbalance? Examination of the common tokenization methods, such as Byte-Pair Encoding (Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016), suggests that subword units induced by tokenization algorithms exhibit similar frequency imbalance to that of full vocabulary. This can be explained by the greedy nature of the vocabulary induction process. Although different methods use different base vocabulary symbols to begin with (i.e., Unicode code points, or bytes), all of the methods construct the vocabulary through iterative merging of the most frequent symbols. As a result, the most frequent units are preserved as words, while the rare words are segmented into subword units. Moreover, the words which are segmented into subword units are

[3] In fact, all three sentences are assigned a negative sentiment, with scores between 97% to 100% by RoBERTa fine-tuned on SST.

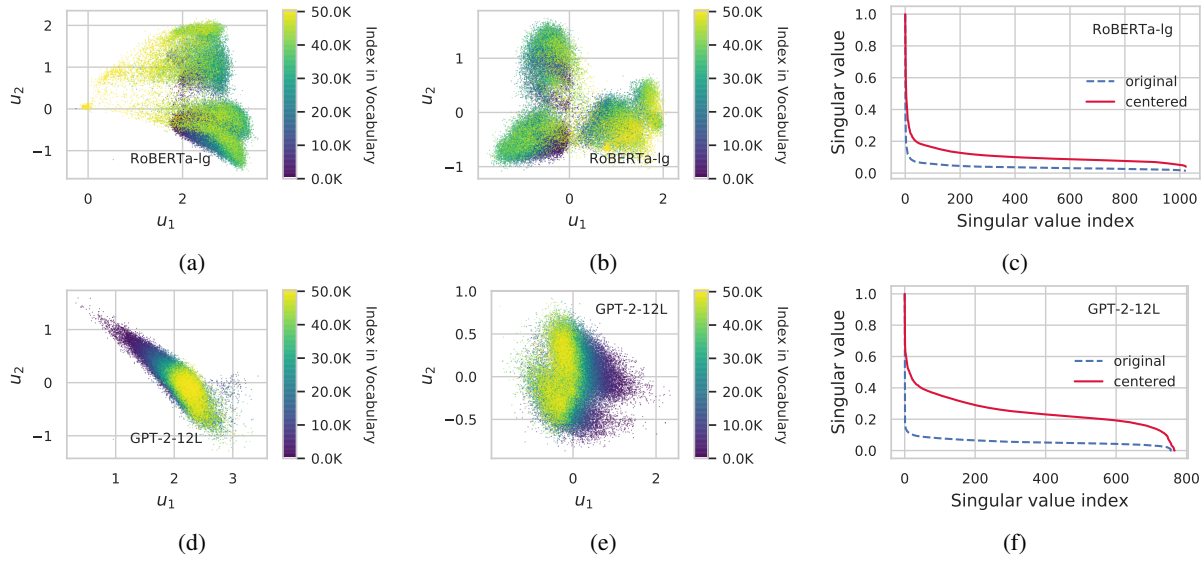


Figure 1: Top: RoBERTa-large. Bottom: GPT-2 (12 layers). (1a, 1d): Word embeddings projected onto first two singular vectors. (1b, 1e) Centered word embeddings projected onto first two singular vectors. (1c, 1f) Singular values of embedding matrix before and after centering. Centering the embedding matrix increases isotropy of embeddings.

infrequent to such a degree that even their combined frequency is orders of magnitude lower than frequency of the most common words.

We confirm this empirically by tokenizing the CNN News corpus (See et al., 2017; Hermann et al., 2015) with WordPiece (used in BERT), revealing that over 30% of the corpus can be accounted for using 13 most frequent tokens, and 50% of the corpus can be accounted for using just 85 tokens. On the other hand, to cover at least 98% of the corpus, nearly 15000 tokens are needed. Therefore, we conclude that the tokens follow approximately Zipfian distribution (Zipf, 1949; Manning and Schütze, 2001) similar to that of full vocabulary. We provide a comparison of frequency distributions of tokens and words based on CNN-News corpus in Appendix B.[4]

3 Learning Language Model

3.1 Autoregressive Language Models

Given a sequence of tokens $\mathbf{w} = [\mathrm{w}_1, ..., \mathrm{w}_N]$ as input, autoregressive (AR) language models assign a probability $p(\mathbf{w})$ to the sequence using factorization $p(\mathbf{w}) = \prod_{t=1}^{N} p(\mathrm{w}_t|\mathrm{w}_{<t})$. Consequently, AR language model is trained by maximizing the likelihood under the forward autoregressive factorization:[5]

$$\max_{\theta} \log p_\theta(\mathbf{w}) = \sum_{t=1}^{N} \log p_\theta(\mathrm{w}_t|\mathbf{w}_{<t}) \quad (1)$$
$$= \sum_{t=1}^{N} \log \frac{\exp\left(\langle \boldsymbol{h}_\theta(\mathbf{w}_{1:t-1})^\top,\ e(\mathrm{w}_t)\rangle\right)}{\sum_{\mathrm{w}'}^{V} \exp\left(\langle \boldsymbol{h}_\theta(\mathbf{w}_{1:t-1})^\top,\ e(\mathrm{w}')\rangle\right)}$$
$$= \sum_{t=1}^{N} \log \mathrm{softmax}\big(\boldsymbol{h}_\theta(\mathbf{w}_{1:t-1})\boldsymbol{W}^\top\big)_{label_t},$$

where $\boldsymbol{h}_\theta(\mathbf{w}_{1:t-1}) \in \mathbb{R}^d$ is the output vector of a model at position t, θ are the model's parameters, $\boldsymbol{W} \in \mathbb{R}^{|V|\times d}$ is the learned embedding matrix, $e(\mathrm{w})$ is a function mapping a token to its representation from the embedding matrix, and $label_t$ is the index of the t-th target token in the vocabulary. To estimate the probability, $\boldsymbol{W}$ maps $\boldsymbol{h}_\theta(\mathbf{w}_{1:t-1})$ to unnormalized scores for every word in the vocabulary V; the scores are subsequently normalized by the softmax to a probability distribution over the vocabulary. In this paper, we focus on neural language models which compute $\boldsymbol{h}_\theta$ using the Transformer architecture, however the mechanisms is generally applicable to other common variants of language models (Mikolov et al., 2010; Sundermeyer et al., 2012; Peters et al., 2018).

[4]The preserved imbalance does not imply that subword tokenization is not beneficial to performance of language systems on rare words. It may mitigate some of the issues as shown in (Sennrich et al., 2016), however recent work demonstrates that it does not solve the problem (Schick and Schütze, 2020b,a).

[5]We omit the bias term in softmax for clarity.

3.2 Masked Language Modeling

Masked Language Modeling (MLM) pretraining objective is to maximize the likelihood of masked tokens conditioned on the (noisy) input sequence. Given a sequence of tokens $\mathbf{w} = [\mathrm{w}_1, ..., \mathrm{w}_N]$, a corrupted version $\hat{\mathbf{w}}$ is constructed by randomly setting a portion of tokens in $\mathbf{w}$ to a special [MASK] symbol. Although MLM estimates the token probabilities of all masked positions, $\bar{\mathbf{w}}$, simultaneously and renders the factorization from Subsection 3.1 no longer applicable, the mechanism used to "unmask" a token differs only slightly from that in AR, specifically:

$$\max_{\theta} \log p_\theta(\bar{\mathbf{w}}|\hat{\mathbf{w}}) \approx \sum_{t=1}^{N} \mathrm{m}_t \log p_\theta(\mathrm{w}_t|\hat{\mathbf{w}}) \quad (2)$$

$$= \sum_{t=1}^{N} \mathrm{m}_t \log \frac{\exp\left(\left\langle \boldsymbol{h}_\theta(\hat{\mathbf{w}})_t^\top,\ e(\hat{\mathrm{w}}_t)\right\rangle\right)}{\sum_{\mathrm{w}'}^{V} \exp\left(\left\langle \boldsymbol{h}_\theta(\hat{\mathbf{w}})_t^\top,\ e(\mathrm{w}')\right\rangle\right)}$$

$$= \sum_{t=1}^{N} \mathrm{m}_t \log \mathrm{softmax}\left(\boldsymbol{h}_\theta(\hat{\mathbf{w}})\boldsymbol{W}^\top\right)_{label_t},$$

where $\mathrm{m}_t = 1$ indicates w_t is masked, and $\boldsymbol{h}_\theta(\hat{\mathbf{w}})_t$ is the output representations computed as function of the full, noisy, input sequence. Note, that the main difference between the equations 1 and 2 is the context used to condition the estimation. Models trained with MLM objective, like BERT and RoBERTa, compute the output vector utilizing bidirectional context through the self-attention mechanism, while the unidirectional models use only the context to the left of the target token. Moreover, only the probabilities of masked words, w_i such that $\mathrm{w}_i \in \bar{\mathrm{w}}$, are estimated.

3.3 Learning Rules

Although the two objectives described above differ in terms of the distribution modeled (Yang et al., 2019), both AR and MLM models rely on the softmax function and cross-entropy loss. Using the notation established above, the cross-entropy loss function for an AR model is optimized by minimizing:

$$J(\theta) = -\mathbb{E}_{\mathbf{w}\sim\mathrm{data}}\left[\log p_\theta(\mathbf{w})\right], \quad (3)$$

and for a MLM model it takes a form of:

$$J(\theta) = -\mathbb{E}_{\mathbf{w}\sim\mathrm{data}}\left[\log p_\theta(\bar{\mathbf{w}}|\hat{\mathbf{w}})\right]. \quad (4)$$

The gradient of the cross-entropy loss with respect to the embedding matrix $\boldsymbol{W}$ is a sum of the gradient flowing through two paths: first one is through the output layer where the embeddings are used to create the targets for the softmax, the second path flows through the encoder stack to the input layer. The gradient flowing through the embedding stack to the input layer is complex, and depends on minute details of a model. Although its contribution is not irrelevant, it is not necessary to illustrate the main point of this section. Thus, we focus on the update rule resulting from the gradient with respect to embeddings in the top layer of a model. For prediction of a token w_t, let $\boldsymbol{h}_\theta$ be the output vector of either AR model (at index $t-1$) or MLM model (at index t), let $y = \mathrm{softmax}(f_t)$, where $f_t = \boldsymbol{h}_\theta \boldsymbol{W}^\top$, and let $\hat{y}$ be the true probability distribution, then:

$$\frac{\partial J_t}{\partial \boldsymbol{W}} = h_\theta(\hat{\mathbf{x}})_t^\top \cdot (y - \hat{y}). \quad (5)$$

The resulting update rule for the embedding matrix is:

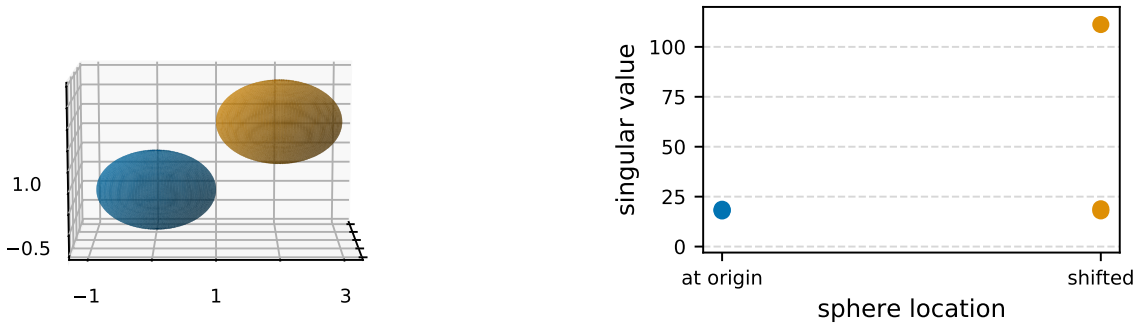
$$\begin{aligned} \boldsymbol{W}' &= \boldsymbol{W} - \eta \cdot (\boldsymbol{h}_\theta^\top \cdot (y - \hat{y})) \\ &= \boldsymbol{W} - \eta \cdot \boldsymbol{h}_\theta^\top y + \eta \cdot \boldsymbol{h}_\theta^\top \hat{y}, \end{aligned} \quad (6)$$

where η be the learning rate. Since $\hat{y}$ is equal to 0 for all the indices except for the index of the target word $\mathbf{w}_t$, all the embeddings will become less similar to the representation produced by a model with the exception of the target word embedding. This leads to what we define as the common enemies effect – target words producing gradients of the same direction for all of the non-target words. As the parameters θ are updated during the optimization process, the $\boldsymbol{h}_\theta$ changes even when the model is provided with the same input. Therefore, the direction of the gradient for the non-target words changes accordingly, but at a particular step the direction of the update is the same for all the non-target words. This is fundamentally different from the conclusion of Gao et al. (2019), who states that there exists a uniformly negative direction such that its minimization yields a nearly optimal solution for rare words' embeddings. We find that the common enemies effect is the most pronounced in the representations of rare words, which are less likely to appear as targets, but it is evident in all embeddings nonetheless.

4 Methods

4.1 Geometry of Embeddings

Previous studies (Gao et al., 2019; Wang et al., 2020) suggest that word embeddings learned by



(a) (b)

Figure 2: A toy illustration of the effect that updates in one direction have on geometry of the representations and their singular values. The singular values in 2b correspond to the spheres of the same color in 2a. As the sphere moves away from the origin, the gap between the singular values of the points sampled from the sphere increases.

Transformer-based language models degenerate and occupy a narrow cone in the embedding space, but instead we find that embeddings simply drift in a common, dominant direction. The conclusions of Gao et al. (2019) are strongly influenced by a rapid decay of singular values of an embedding matrix, however, a rapid decay of singular values is not a sufficient condition to reach such conclusions.

In fact, points sampled from a 3D sphere satisfy the condition given above. As at first glance this is not entirely obvious, we provide a toy example in Figure 2 that illustrates why embeddings appear as a cone when projected to a low dimensional space. We sample points at random from two spheres, one centered at the origin and one shifted away from the origin (Figure 2a), and perform Singular Value Decomposition on the two sets of samples. When the sphere moves away from the origin, the difference between the two singular values increases (Figure 2b).

Similarly, the projection of uncentered embeddings (see Figures 1a and 1d) appears as a cone, but when embeddings are centered around origin (Figures 1b, 1e), the shape of their projection changes to resemble a sphere more than a cone; that is simply removing the mean vector $\boldsymbol{\mu}$ of an embedding matrix $\boldsymbol{W}$, where $\boldsymbol{\mu} = \sum_{\mathrm{w} \in \boldsymbol{W}} e(\mathrm{w}) \;/\; |V|$, increases the isotropy of embeddings. Optimization of a neural language model is certainly more complex than our toy example. Most of all, the common enemy effect is not uniform; the amount by which each vector moves in the most dominant direction depends on many factors, among others the size of the training corpus, the diversity of the training corpus, or whether static (BERT) or dynamic (RoBERTa) masking is used. In a more general sense, the magnitude of the gradient with respect to a word vector depends on the value in the logit corresponding to that word, hence the shift will not be uniform.

4.2 Unused Tokens and Rare Words

We hypothesize that as rare words drift in common direction, their embeddings become less discriminative than embeddings of frequent words. BERT's vocabulary provides a unique opportunity to investigate the contribution of the same direction gradients to embeddings of particular words. There are 994 special *unused* tokens in BERT's vocabulary that were not used as inputs or targets during pretraining, thus all the updates to their representations were in the directions opposite to output vectors. As shown in Figure 3, we observe that cosine similarity between the unused tokens and other tokens increases as the frequency decreases. The average cosine similarity between unused words and tokens in indices [28500-29500][6] is 0.63. In comparison the unused tokens have cosine similarity of 0.27 with tokens in indices [2000-3000] (most frequent tokens, i.e., "to") but the similarity goes up rapidly for tokens other than the most frequent ones.[7] Schick and Schütze (2019), evaluate BERT and RoBERTa on a dataset explicitly measuring the ability of MLM models to "unmask" words of different frequencies, and report that both models struggle to "unmask" rare words. Results presented in this section provide an explanation of this behavior and confirm that embeddings of the rare tokens

[6] Although frequency depends on a corpus, in general higher index implies lower frequency due to the way BERT's vocabulary is constructed.

[7] We observe a similar pattern in RoBERTa using the last 1000 words in its vocabulary in place of the *unused* tokens.

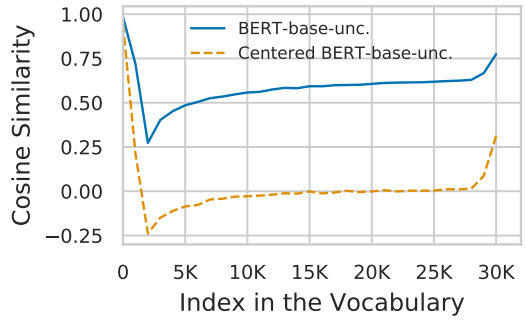


Figure 3: Cosine similarity between the [unused] tokens and words in vocabulary of BERT-base-case grouped into bins of 1000 (i.e., [1000:1999]).

are most affected by the common enemies effect.

5 Experiments

We validate our theoretical analysis through a series of experiments on geometric properties of non-contextualized embeddings.

5.1 Isotropy

Although centering an embedding matrix results in a more desirable spectral distribution, tokens of comparable frequency tend to remain clustered in the embedding space, as shown in Fig 1. Therefore, we empirically test how much actual gain in terms of isotropy is obtained in embeddings of the tested models by removing the shared direction. Moreover, Mu and Viswanath (2018) show that the top principal components in skip-gram embeddings (Mikolov et al., 2013a) correspond to frequency of words and demonstrate that such frequency bias can be mitigated by removing the top principal components of an embedding matrix. We evaluate the effectiveness of this approach on embeddings from Transformer-based models. We use BERT, RoBERTa, and GPT-2 in different sizes in our experiments.

Setup: We measure the initial isotropy of embeddings in each of the models, and the isotropy after removing the mean vector $\boldsymbol{\mu} = \sum_{\boldsymbol{w} \in \boldsymbol{W}} \boldsymbol{w}/|V|$ from each row of an embedding matrix $\boldsymbol{W}$, yielding $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \boldsymbol{\mu}$. Next, we use a slightly modified approach of Mu and Viswanath (2018), and remove D top principal components from each model's embedding matrix to obtain highly isotropic representations. Finally, we evaluate whether increasing isotropy of embeddings from Transformer-based models can improve performance on standard embedding benchmarks.

Definitions: To measure isotropy, we use the partition function defined in (Arora et al., 2016),

$$\mathrm{Z}(\boldsymbol{c}) = \sum_{\mathrm{w} \in V} \exp(\boldsymbol{c}^\top e(\mathrm{w})), \tag{7}$$

where $e(\mathrm{w})$ maps a word w to its embedding and $\boldsymbol{c}$ is a unit vector. For vectors to be isotropic, the value of $\mathrm{Z}(\boldsymbol{c})$ should be approximately constant, according to Lemma 2.1 in (Arora et al., 2016). Based on this property, we empirically measure the isotropy of an embedding matrix $\boldsymbol{W}$ using:

$$\mathrm{I}(\boldsymbol{W}) = \frac{\min_{\boldsymbol{c} \in \boldsymbol{X}} \mathrm{Z}(\boldsymbol{c})}{\max_{\boldsymbol{c} \in \boldsymbol{X}} \mathrm{Z}(\boldsymbol{c})}, \tag{8}$$

where $\mathrm{I}(\boldsymbol{W}) \in [0, 1]$. We follow the standard approach and define $\boldsymbol{X}$ to be the set of eigenvectors of $\boldsymbol{W}^\top \boldsymbol{W}$ (Mu and Viswanath, 2018; Wang et al., 2020). We remove the top principal components using a modified version of the post-processing method proposed by Mu and Viswanath (2018):

$$\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i - \frac{1}{|V|} \sum_{j=1}^{V} \boldsymbol{W}_j \tag{9}$$

$$\boldsymbol{U} = \mathrm{PCA}(\tilde{\boldsymbol{W}}) \tag{10}$$

$$\hat{\boldsymbol{W}}_i = \tilde{\boldsymbol{W}}_i - \sum_{j=1}^{\mathrm{D}} (\boldsymbol{U}_j^\top \tilde{\boldsymbol{W}}_i) \boldsymbol{U}_j, \tag{11}$$

where $\boldsymbol{W}$ is the embedding matrix, $\hat{\boldsymbol{W}}$ is the post-processed embedding matrix, and D is the number of principal components removed from the original matrix. Mu and Viswanath (2018) use $\boldsymbol{W}$ instead of $\tilde{\boldsymbol{W}}$ in the term $(\boldsymbol{U}_j^\top \tilde{\boldsymbol{W}}_i)\boldsymbol{U}_j$ in eq. 11, but we find the centered version of $\boldsymbol{W}$ to be more effective. Following Mu and Viswanath (2018), we set $\mathrm{D} = \lceil d/100 \rceil$, where d is the dimensionality of a model.

5.2 Embedding Benchmarks

Setup: We evaluate each model's embedding's performance on common benchmarks for word similarity and relatedness before and after post-processing. We use the following data sets:

- **SimLex-999** (Hill et al., 2015) - measures similarity, rather than relatedness or association.
- **MEN Test Collection** (Bruni et al., 2014) - measures the relatedness of words.
- **WordSim353** (Agirre et al., 2009) - consists of two parts, one measures similarity, and the other measures relatedness of words.

Model	$\mathrm{I}(\boldsymbol{W})$	$\mathrm{I}(\boldsymbol{W}_c)$	$\mathrm{I}(\boldsymbol{W}_r)$	$\mathrm{avg}(\|\|\mathrm{e(w)}\|\|_2)$	$\|\|\boldsymbol{\mu}\|\|_2$	$\|\|\boldsymbol{\mu}\|\|_2/\mathrm{avg}(\|\|\mathrm{e(w)}\|\|_2)$
BERT-base-uncased	0.39	**0.98**	**0.998**	1.40	0.94	0.67
BERT-base-cased	**0.59**	**0.98**	0.996	1.29	0.50	0.39
BERT-large-uncased	0.44	0.97	0.997	1.45	0.80	0.55
BERT-large-cased	0.52	0.96	0.995	1.53	0.65	0.42
RoBERTa-base	0.50	0.87	0.959	3.65	0.57	0.16
RoBERTa-large	0.07	0.64	0.956	4.36	2.53	0.58
GPT-2 (12 layers)	0.12	0.91	0.969	3.96	2.05	0.52
GPT-2 (24 layers)	0.52	0.95	0.981	3.68	2.04	0.55

Table 1: Isotropy, $\mathrm{I}(\boldsymbol{W}) \in [0, 1]$, of embeddings from various language models. Centering an embedding matrix yields nearly perfectly isotropic embeddings in most of the tested models. $\boldsymbol{W}_c$ stands for a centered matrix, $\boldsymbol{W}_r$ stands for an embedding matrix with $\lceil d/100 \rceil$ top principal components removed. $||\boldsymbol{\mu}||_2/\mathrm{avg}(||\mathrm{e(w)}||_2$ is the ratio of the L_2 norm of the mean vector, $\boldsymbol{\mu}$, to the average of the L_2 norms of word embeddings.

- **Stanford Rare Words (RW)** (Luong et al., 2013) - measures similarity of words. In this dataset at least one word in each pair is a rare word.

The data sets are designed to measure embeddings' ability to reflect semantic relations. The performance on the data sets is measured by the correlation between the similarities of the representations and the human scores. We filter out samples consisting of subword units. Although this results in different test sets for different models, our goal is not to compare different models' performance but to validate the benefits of increased isotropy of embeddings. We score relations with both cosine similarity and inner product.

Schakel and Wilson (2015) show that vectors of more frequent words tend to have smaller norms, which was confirmed for BERT by Podkorytov et al. (2020). As the longer vectors of rare words are most affected by common enemies effect (see Section 4.2), we evaluate a "scaled-centering" method to account for that.

Specifically, we first compute the mean vector of embeddings normalized to unit length $\hat{\boldsymbol{\mu}} = \sum_{\mathrm{w} \in \boldsymbol{W}} \frac{e(\mathrm{w})}{||e(\mathrm{w})||_2} \; / \; |V|$. Then we scale the mean vector by the norm of each word embedding before subtracting it, $e(\mathrm{w})' = e(\mathrm{w}) - ||e(\mathrm{w})||_2 \; \hat{\boldsymbol{\mu}}$.

5.3 Results

Isotropy: We find that merely removing the mean vector is enough for most models to reach nearly perfect isotropy. The results are in Table 1. The only exception is RoBERTa-large, which had the lowest initial isotropy. Interestingly, Schick and Schütze (2020a) show that RoBERTa-large outperforms BERT models on tasks designed explicitly for rare words. Moreover, according to common leaderboards (Wang et al., 2019b,a), RoBERTa performs best on downstream tasks among the models we analyzed.

We stress that the $\mathrm{I}(\boldsymbol{W})$ is an approximation of the degree of isotropy, and should be treated as such when interpreting its relation to downstream performance. The idea of the partition function $Z(\boldsymbol{c})$ states that it's value should be constant for any vector $\boldsymbol{c}$ (Arora et al., 2016; Mu and Viswanath, 2018). As there is no closed-form solution for $\min_{\boldsymbol{c} \in \boldsymbol{X}}$ and $\max_{\boldsymbol{c} \in \boldsymbol{X}}$, a set of eigenvectors of $\boldsymbol{W}^\top \boldsymbol{W}$ has been used as $\boldsymbol{X}$ in previous studies to approximate the isotropy (e.g., Mu and Viswanath, 2018; Wang et al., 2020). The vectors in $\boldsymbol{X}$, however, cannot be considered principal components of $\boldsymbol{W}$, unless the matrix $\boldsymbol{W}$ has been centered. Pearson (1901) states that unless the mean of the data has been subtracted, the best fitting hyperplane would pass through the origin and not through the centroid. Indeed, for RoBERTa-large, the cosine similarity between the top eigenvector of $\boldsymbol{W}^\top \boldsymbol{W}$ and the mean vector is 0.99.

Additionally, as the volume of a cube in $\mathbb{R}^n$ grows exponentially with n, it may be sufficient for the embeddings to be isotropic around a point lying on a lower dimensional subspace to retain the desired separation. In fact, embeddings from RoBERTa-large have an average pairwise cosine similarity of 0.33 (angle of $70.7°$).

We speculate that a longer pretraining of RoBERTa compared to BERT results in a more significant shift of the embeddings in the dominating directions. Simultaneously, a larger pretraining corpus and a dynamic masking scheme used in RoBERTa may result in a more diverse set of shift directions. We leave this line of research for future studies.

Moreover, Mu and Viswanath (2018) demon-

Model	CosSim	$\langle \cdot, \cdot \rangle$
BERT-base-cased	62.29 (+0.00)	60.91 (+0.00)
+ Centered	60.44 (−1.85)	60.08 (−0.83)
+ Centered-Scaled	62.32 (+0.03)	62.41 (+1.50)
+ Post-Process	65.57 (+3.28)	66.10 (+5.19)
RoBERTa-base	66.81 (+0.00)	66.01 (+0.00)
+ Centered	66.85 (+0.04)	67.03 (+1.02)
+ Centered-Scaled	67.02 (+0.21)	66.95 (+0.94)
+ Post-Process	66.87 (+0.06)	67.15 (+1.14)
GPT-2-small	64.71 (+0.00)	59.45 (+0.00)
+ Centered	64.95 (+0.24)	66.04 (+6.59)
+ Centered-Scaled	66.57 (+1.86)	67.32 (+7.87)
+ Post-Process	67.85 (+)3.14	67.67 (+8.22)

Model	CosSim	$\langle \cdot, \cdot \rangle$
BERT-large-cased	61.90 (+0.00)	59.42 (+0.00)
+ Centered	58.66 (−3.24)	57.99 (−1.43)
+ Centered-Scaled	61.18 (−0.72)	61.05 (+1.63)
+ Post-Process	65.72 (+3.82)	65.89 (+6.47)
RoBERTa-large	61.29 (+0.00)	44.79 (+0.00)
+ Centered	64.09 (+2.80)	63.78 (+18.99)
+ Centered-Scaled	65.49 (+4.20)	64.16 (+19.37)
+ Post-Process	64.38 (+3.09)	65.17 (+20.38)
GPT-2-medium	65.53 (+0.00)	59.20 (+0.00)
+ Centered	66.83 (+1.30)	67.90 (+8.70)
+ Centered-Scaled	67.95 (+2.42)	68.22 (+9.02)
+ Post-Process	68.05 (+2.52)	67.81 (+8.61)

Table 2: Average performance (Pearson's $r \times 100$) of the models on the non-contextual benchmarks (SimLex-999, MEN, WordSim353, Stanford Rare Words). Centered stands for embedding matrix centered at origin; Centered-Scaled stands for embedding matrix with mean direction, scaled by the norm of each word embedding, subtracted; Post-Process corresponds to the method defined in eq. 11. Results from different models are not directly comparable due to different tokenization. Best results for each model are underlined. In general, increased isotropy results in increased performance. The improvement of RoBERTa-large is more significant as its initial isotropy is lower. Specific results for each benchmark can be found in Appendix C.

strate that neural language models are capable of learning to remove the mean vector. We leave the question whether Transformer-based language models perform an implicit representation centering operation to future research.

Embedding Benchmarks: We present our results on common benchmarks for word similarity and relatedness in Table 2. We report average scores from all tasks. The results on individual data sets are available in Appendix C. We observe that removing the mean vector, and consequently increasing the isotropy of embeddings, consistently improves the performance across all models, except for the most isotropic BERT-cased models. Furthermore, results in Table 2 demonstrate that "scaled-centering" is more effective than simple mean subtraction, and nearly as effective as the more expensive post-processing method. The only case in which "scaled-centering" does not improve performance is BERT-large-cased with cosine similarity as a scoring function.

Performance gains are more pronounced when inner-product is used as a scoring function, regardless of the model or processing method used. Although, initially cosine-similarity yields better results, especially for embeddings with greater L_2 norms, mean subtraction is sufficient to close the gap in all but two models (BERT-cased models).[8]

[8]Visualization of distributions of L_2 norms of embeddings from the analyzed models is available in Appendix C.

6 Discussion

There has been a body of literature demonstrating substantial benefits of improved quality of word embeddings on downstream performance (e.g., Mu and Viswanath, 2018; Wang et al., 2019c; Gao et al., 2019; Wang et al., 2020; Schick and Schütze, 2020a). In particular, Gao et al. (2019) propose to add a cosine similarity regularization to the cross-entropy loss to increase the aperture of the cone in which embeddings are distributed, and report improved performance on machine translation and language modeling. It is straightforward to demonstrate that the cosine regularization proposed by Gao et al. (2019) is equivalent to minimizing the squared norm of the mean direction of embeddings, hence constraining the most significant drift direction. We provide the derivation of the equivalence in Appendix A.

Large-margin classification has been studied extensively, both in NLP (Wang et al., 2019c) and machine learning in general (Weston and Watkins, 1999; Tsochantaridis et al., 2005). As substantial shared components of embeddings will lead to a decreased classification margin in the output softmax layer, our work offers explanation for the fragility of pretrained language models reported in the literature (e.g., Schick and Schütze 2019, 2020a).

Our analyses show clearly that shifting of the embeddings in the embedding space is due to the dynamic interactions between the representations and the embedding vectors. As the embeddings

become more similar, the resulting representations become closer, creating a positive feedback mechanism for the representations to drift collectively. In addition, while isotropy of representations is desirable and has an overall positive impact on performance, the relationships between isotropy and performance in Table 1 and Table 2 suggest that the role of isotropy in model performance needs to be further analyzed. The dynamics of the interactions are being further investigated to pinpoint the root cause and their relationship with the model's performance.

7 Related Work

Gao et al. (2019) present an insightful derivation of uniformly negative gradients for nonapparent words and formulate the optimization of rare words as an α-strongly convex problem but make strong assumptions that the embedding matrix is learned after all other parameters of the model are well-optimized and fixed, which is not the case in practice. We do not make such assumptions, providing a more realistic explanation for the learning process. Wang et al. (2020) propose to reparametrize the embedding matrix using SVD and propose directly controlling the decay rate of singular values. Our paper's purpose is inherently different from that of Wang et al. (2020); we recognize that the fundamental understanding of the problem is missing and provide an explanation for the observations made in previous studies. Another line of work focuses on limitations of the softmax. Yang et al. (2018) suggest that softmax does not have sufficient capacity to model the complexity of language. Zhang et al. (2019) analyze the skip-gram model to show that optimization based on cross-entropy loss and softmax resembles competitive learning in which words compete among each other for the context vector. This idea is closely related to the common enemies effect reported in this paper, however, skip-gram seems to mitigate this through negative sampling (Mikolov et al., 2013b) but similar approaches do not seem to help Transformer pre-training (Clark et al., 2020).

A considerable effort has been made to improve performance of language systems on rare words, but the focus has been on either injecting subword information in non-contextual representations (Luong et al., 2013; Lazaridou et al., 2017; Pinter et al., 2017; Bojanowski et al., 2017), replacing rare words' representations through exploiting their context (Khodak et al., 2018; Liu et al., 2019a), or both (Schick and Schütze, 2019, 2020a). In comparison, we strive to provide an explanation of the underlying problem, which is necessary to render such post-hoc fixes no longer necessary.

8 Conclusion

We find that the embeddings learned by GPT-2, BERT, and RoBERTa do not degenerate into a narrow cone, as has been suggested in the past, but instead drift in one shared direction. We recognize that target words produce gradients in the same direction for all the non-target words at each training step. Combined with the unbalanced distribution of word frequencies, any two words' embeddings will be repeatedly updated with gradients of the same direction. As such updates accumulate, the embeddings drift and share common components. Our experiments show that simply centering the embeddings restores a nearly perfectly isotropic distribution of tested models' embeddings and simultaneously improves embeddings' ability to reflect semantic relations. This understanding of the learning process dynamics opens exciting avenues for future work, such as improving the most affected embeddings of rare words and formulation of more computationally efficient training objectives.

References

Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *HLT-NAACL*.

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

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. In *J. Mach. Learn. Res.*

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with

subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *EMNLP/IJCNLP*.

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*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41:665–695.

Hakan Inan, Khashayar Khosravi, and R. Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. *ArXiv*, abs/1611.01462.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Melbourne, Australia. Association for Computational Linguistics.

A. Lazaridou, Marco Marelli, and M. Baroni. 2017. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive science*, 41 Suppl 4:677–705.

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2019a. Second-order contexts from lexical substitutes for few-shot learning of word representations. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 61–67, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, Melbourne, Australia. 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. 2019b. Roberta: A robustly optimized bert pretraining approach. In *arXiv:1907.11692*.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*.

Christopher D. Manning and Hinrich Schütze. 2001. Foundations of statistical natural language processing. In *SGMD*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

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

Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. In *Philosophical Magazine*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

S. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21:1112–1130.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112, Copenhagen, Denmark. Association for Computational Linguistics.

Maksim Podkorytov, Daniel Bis, Jinglun Cai, Kobra Amirizirtol, and X. Liu. 2020. Effects of architecture and training on embedding geometry and feature discriminability in bert. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Adriaan MJ Schakel and Benjamin J Wilson. 2015. Measuring word significance using distributed representations of words. In *arXiv:1508.02297*.

Timo Schick and Hinrich Schütze. 2019. Attentive mimicking: Better word embeddings by attending to informative contexts. 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 489–494, Minneapolis, Minnesota. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020a. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020b. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *AAAI*.

M. Schuster and K. Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *INTERSPEECH*.

Ioannis Tsochantaridis, T. Joachims, Thomas Hofmann, and Y. Altun. 2005. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019c. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565, Long Beach, California, USA. PMLR.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

J. Weston and C. Watkins. 1999. Support vector machines for multi-class pattern recognition. In *ESANN*.

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. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's

neural machine translation system: Bridging the gap between human and machine translation. In *arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Canlin Zhang, Xiuwen Liu, and Daniel Bis. 2019. An analysis on the learning rules of the skip-gram model. *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

George Kingsley Zipf. 1949. Human behaviour and the principle of least effort: an introduction to human ecology.

A Cosine Regularization as Mean Direction Minimization

In this section we show the equivalence of Cosine Regularization and mean direction minimization.

$$\text{CosReg} = \frac{1}{N^2}\sum_{i}^{N}\sum_{j\neq i}^{N}\hat{\boldsymbol{w}}_i^\top\hat{\boldsymbol{w}}_j \tag{12}$$

$$= \frac{1}{N^2}\big(\sum_{i}^{N}\hat{\boldsymbol{w}}_i\big)^\top\big(\sum_{i}^{N}\hat{\boldsymbol{w}}_i\big) - \text{N} \tag{13}$$

$$= \frac{1}{N^2}||\sum_{i}^{N}\hat{\boldsymbol{w}}_i||_2^2 - \text{N}, \tag{14}$$

as N is a constant, minimizing $\frac{1}{N}||\sum_i^N \hat{\boldsymbol{w}}_i||_2$ is equivalent to minimizing the CosReg term.

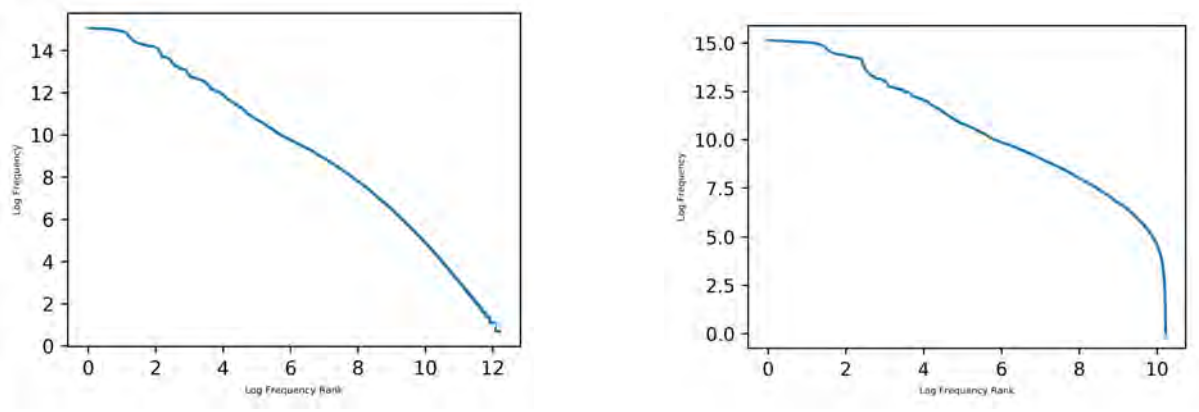
B Token Frequency Validation

Figure 4 validates that the word frequency imbalance is preserved in a corpus tokenized with WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016).

C Additional Experimental Results

In Table 3, we provide expanded results on the embedding benchmarks (see Section 5.2 for details).

Our experiments reveal a *negative* 0.61 correlation between the average norm of the embedding vectors and their isotropy. Additionally, the ratio of the L_2 norm of the mean vector to the average of the L_2 norms of embeddings tends to be larger for less isotropic embeddings. Figure 5 shows distributions of L_2 norms of embeddings from the models studied in this paper. Moreover, Figure 6 compares the effect of centering on the L_2 norms of embeddings from the BERT-large-cased and RoBERTa-large. We leave the relationship between the norms of the vectors, their isotropy, and the pretraining details (e.g., corpus size, number of training steps, weight decay) for future studies.



(a) Words (b) Tokens

Figure 4: Comparison of word and token frequencies on CNN-DailyMail corpus.

Model	**MEN** $\langle\cdot,\cdot\rangle$/cos	**RW** $\langle\cdot,\cdot\rangle$/cos	**Simlex-999** $\langle\cdot,\cdot\rangle$/cos	**WordSim-Sim** $\langle\cdot,\cdot\rangle$/cos	**WordSim-Rel** $\langle\cdot,\cdot\rangle$/cos
Bert-base-cased	65.53 / 67.07	61.22 / 63.72	52.52 / 52.37	75.03 / 75.57	50.27 / 52.70
+ Centered-Scaled	69.32 / 68.36	63.57 / 63.94	47.34 / 47.96	78.58 / 77.36	53.23 / 53.96
+ Post-Process	72.20 / 70.91	67.71 / 68.20	53.60 / 52.53	79.38 / 78.15	57.63 / 58.04
Bert-large-cased	64.64 / 67.20	58.94 / 61.89	52.68 / 52.91	73.44 / 76.32	47.40 / 51.16
+ Centered-Scaled	68.63 / 67.69	61.67 / 62.19	48.66 / 48.53	77.47 / 77.45	48.81 / 50.02
+ Post-Process	72.12 / 71.07	67.22 / 68.07	55.41 / 53.77	78.80 / 78.88	55.90 / 56.79
GPT-2-small	65.55 / 71.16	58.08 / 63.99	50.43 / 51.45	72.98 / 78.32	50.19 / 58.63
+ Centered-Scaled	75.55 / 74.10	64.05 / 64.94	51.60 / 50.36	81.41 / 80.61	63.98 / 62.86
+ Post-Process	76.00 / 75.34	65.30 / 66.56	53.74 / 53.15	80.70 / 80.79	62.62 / 63.40
GPT-2-medium	65.01 / 71.55	58.55 / 65.22	51.13 / 52.51	72.67 / 78.48	48.65 / 59.90
+ Centered-Scaled	76.10 / 75.02	64.14 / 65.34	54.22 / 53.30	80.85 / 80.56	65.79 / 65.51
+ Post-Process	75.92 / 75.25	64.80 / 66.11	54.65 / 54.19	79.99 / 80.25	63.71 / 64.43
RoBERTa-base	72.83 / 72.70	64.72 / 66.61	55.17 / 54.84	78.00 / 78.62	59.31 / 61.27
+ Centered-Scaled	74.18 / 73.33	65.38 / 66.35	54.16 / 53.77	79.35 / 79.10	61.66 / 62.53
+ Post-Process	74.22 / 73.13	65.45 / 66.11	53.87 / 53.35	79.64 / 79.08	62.58 / 62.70
RoBERTa-large	42.92 / 63.44	49.72 / 64.31	45.75 / 54.54	58.09 / 73.46	27.47 / 50.69
+ Centered-Scaled	70.55 / 70.95	63.09 / 65.63	55.24 / 54.18	75.72 / 77.34	56.19 / 59.35
+ Post-Process	72.01 / 70.55	63.11 / 63.64	52.48 / 50.93	77.87 / 77.09	60.36 / 59.71

Table 3: Performance (Pearson's $r \times 100$) of the models on the non-contextual benchmarks. Centered-Scaled stands for embedding matrix with mean direction, scaled by the norm of each word embedding, subtracted; Post-Process corresponds to the method defined in eq. 11. Results from different models are not directly comparable due to different tokenization.

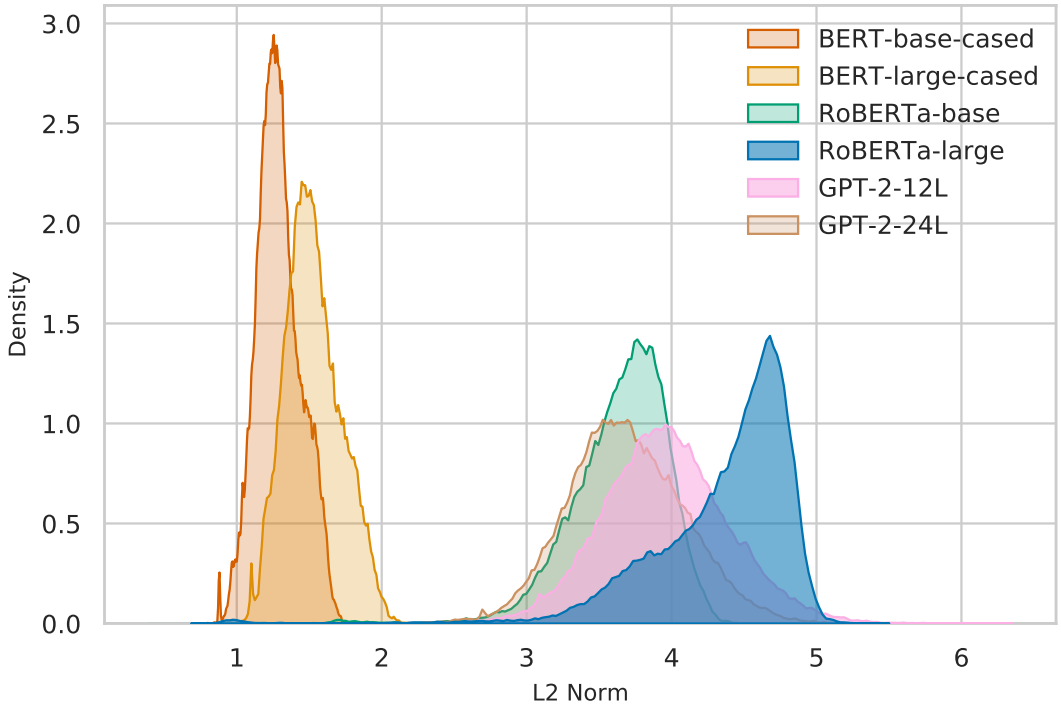
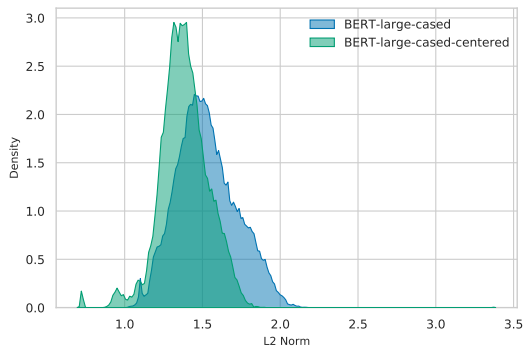
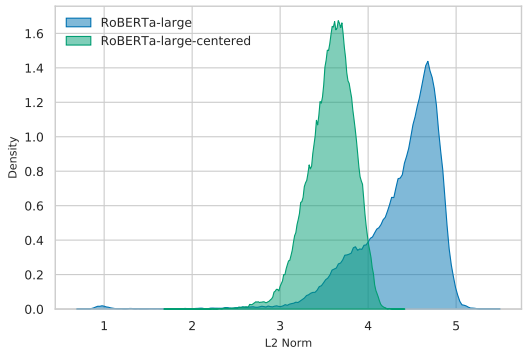


Figure 5: Kernel density estimation of the L_2 norms of embeddings from different models. Models trained on larger corpora and with an increased number of pretraining steps exhibit larger embedding norms.



(a) BERT-large-cased



(b) RoBERTa-large

Figure 6: The effect of mean subtraction on the distribution of L_2 norms of embeddings in BERT-large-cased and RoBERTa-large.