How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings

Kawin Ethayarajh[*]
Stanford University
`kawin@stanford.edu`

Abstract

Replacing static word embeddings with contextualized word representations has yielded significant improvements on many NLP tasks. However, just how contextual are the contextualized representations produced by models such as ELMo and BERT? Are there infinitely many context-specific representations for each word, or are words essentially assigned one of a finite number of word-sense representations? For one, we find that the contextualized representations of all words are not isotropic in any layer of the contextualizing model. While representations of the same word in different contexts still have a greater cosine similarity than those of two different words, this self-similarity is much lower in upper layers. This suggests that upper layers of contextualizing models produce more context-specific representations, much like how upper layers of LSTMs produce more task-specific representations. In all layers of ELMo, BERT, and GPT-2, on average, less than 5% of the variance in a word's contextualized representations can be explained by a static embedding for that word, providing some justification for the success of contextualized representations.

1 Introduction

The application of deep learning methods to NLP is made possible by representing words as vectors in a low-dimensional continuous space. Traditionally, these word embeddings were *static*: each word had a single vector, regardless of context (Mikolov et al., 2013a; Pennington et al., 2014). This posed several problems, most notably that all senses of a polysemous word had to share the same representation. More recent work, namely deep neural language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018),

---

*Work partly done at the University of Toronto.

have successfully created *contextualized word representations*, word vectors that are sensitive to the context in which they appear. Replacing static embeddings with contextualized representations has yielded significant improvements on a diverse array of NLP tasks, ranging from question-answering to coreference resolution.

The success of contextualized word representations suggests that despite being trained with only a language modelling task, they learn highly transferable and task-agnostic properties of language. In fact, linear probing models trained on frozen contextualized representations can predict linguistic properties of words (e.g., part-of-speech tags) almost as well as state-of-the-art models (Liu et al., 2019a; Hewitt and Manning, 2019). Still, these representations remain poorly understood. For one, just how contextual are these contextualized word representations? Are there infinitely many context-specific representations that BERT and ELMo can assign to each word, or are words essentially assigned one of a finite number of word-sense representations?

We answer this question by studying the geometry of the representation space for each layer of ELMo, BERT, and GPT-2. Our analysis yields some surprising findings:

1. In all layers of all three models, the contextualized word representations of all words are not isotropic: they are not uniformly distributed with respect to direction. Instead, they are *anisotropic*, occupying a narrow cone in the vector space. The anisotropy in GPT-2's last layer is so extreme that two random words will on average have almost perfect cosine similarity! Given that isotropy has both theoretical and empirical benefits for static embeddings (Mu et al., 2018), the extent of anisotropy in contextualized represen-

tations is surprising.

2. Occurrences of the same word in different contexts have non-identical vector representations. Where vector similarity is defined as cosine similarity, these representations are more dissimilar to each other in upper layers. This suggests that, much like how upper layers of LSTMs produce more task-specific representations (Liu et al., 2019a), upper layers of contextualizing models produce more context-specific representations.

3. Context-specificity manifests very differently in ELMo, BERT, and GPT-2. In ELMo, representations of words in the same sentence grow more similar to each other as context-specificity increases in upper layers; in BERT, they become more dissimilar to each other in upper layers but are still more similar than randomly sampled words are on average; in GPT-2, however, words in the same sentence are no more similar to each other than two randomly chosen words.

4. After adjusting for the effect of anisotropy, on average, less than 5% of the variance in a word's contextualized representations can be explained by their first principal component. This holds across all layers of all models. This suggests that contextualized representations do not correspond to a finite number of word-sense representations, and even in the best possible scenario, static embeddings would be a poor replacement for contextualized ones. Still, static embeddings created by taking the first principal component of a word's contextualized representations outperform GloVe and FastText embeddings on many word vector benchmarks.

These insights help justify why the use of contextualized representations has led to such significant improvements on many NLP tasks.

2 Related Work

Static Word Embeddings Skip-gram with negative sampling (SGNS) (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are among the best known models for generating static word embeddings. Though they learn embeddings iteratively in practice, it has been proven that in theory, they both implicitly factorize a word-context matrix containing a co-occurrence statistic (Levy and Goldberg, 2014a,b). Because they create a single representation for each word, a notable problem with static word embeddings is that all senses of a polysemous word must share a single vector.

Contextualized Word Representations Given the limitations of static word embeddings, recent work has tried to create context-sensitive word representations. ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and GPT-2 (Radford et al., 2019) are deep neural language models that are fine-tuned to create models for a wide range of downstream NLP tasks. Their internal representations of words are called *contextualized word representations* because they are a function of the entire input sentence. The success of this approach suggests that these representations capture highly transferable and task-agnostic properties of language (Liu et al., 2019a).

ELMo creates contextualized representations of each token by concatenating the internal states of a 2-layer biLSTM trained on a bidirectional language modelling task (Peters et al., 2018). In contrast, BERT and GPT-2 are bi-directional and uni-directional transformer-based language models respectively. Each transformer layer of 12-layer BERT (base, cased) and 12-layer GPT-2 creates a contextualized representation of each token by attending to different parts of the input sentence (Devlin et al., 2018; Radford et al., 2019). BERT – and subsequent iterations on BERT (Liu et al., 2019b; Yang et al., 2019) – have achieved state-of-the-art performance on various downstream NLP tasks, ranging from question-answering to sentiment analysis.

Probing Tasks Prior analysis of contextualized word representations has largely been restricted to probing tasks (Tenney et al., 2019; Hewitt and Manning, 2019). This involves training linear models to predict syntactic (e.g., part-of-speech tag) and semantic (e.g., word relation) properties of words. Probing models are based on the premise that if a simple linear model can be trained to accurately predict a linguistic property, then the representations implicitly encode this information to begin with. While these analyses have found that contextualized representations encode semantic and syntactic information, they cannot answer *how contextual* these representations are, and to

what extent they can be replaced with static word embeddings, if at all. Our work in this paper is thus markedly different from most dissections of contextualized representations. It is more similar to Mimno and Thompson (2017), which studied the geometry of static word embedding spaces.

3 Approach

3.1 Contextualizing Models

The contextualizing models we study in this paper are ELMo, BERT, and GPT-2[1]. We choose the base cased version of BERT because it is most comparable to GPT-2 with respect to number of layers and dimensionality. The models we work with are all pre-trained on their respective language modelling tasks. Although ELMo, BERT, and GPT-2 have 2, 12, and 12 hidden layers respectively, we also include the input layer of each contextualizing model as its 0th layer. This is because the 0th layer is not contextualized, making it a useful baseline against which to compare the contextualization done by subsequent layers.

3.2 Data

To analyze contextualized word representations, we need input sentences to feed into our pre-trained models. Our input data come from the SemEval Semantic Textual Similarity tasks from years 2012 - 2016 (Agirre et al., 2012, 2013, 2014, 2015). We use these datasets because they contain sentences in which the same words appear in different contexts. For example, the word 'dog' appears in *"A panda dog is running on the road."* and *"A dog is trying to get bacon off his back."* If a model generated the same representation for 'dog' in both these sentences, we could infer that there was no contextualization; conversely, if the two representations were different, we could infer that they were contextualized to some extent. Using these datasets, we map words to the list of sentences they appear in and their index within these sentences. We do not consider words that appear in less than 5 unique contexts in our analysis.

3.3 Measures of Contextuality

We measure how contextual a word representation is using three different metrics: *self-similarity*, *intra-sentence similarity*, and *maximum explainable variance*.

Definition 1 Let w be a word that appears in sentences $\{s_1, ..., s_n\}$ at indices $\{i_1, ..., i_n\}$ respectively, such that $w = s_1[i_1] = ... = s_n[i_n]$. Let $f_\ell(s, i)$ be a function that maps $s[i]$ to its representation in layer ℓ of model f. The *self similarity* of w in layer ℓ is

$$SelfSim_\ell(w) = \frac{1}{n^2 - n} \sum_j \sum_{k \neq j} \cos(f_\ell(s_j, i_j), f_\ell(s_k, i_k)) \tag{1}$$

where cos denotes the cosine similarity. In other words, the *self-similarity* of a word w in layer ℓ is the average cosine similarity between its contextualized representations across its n unique contexts. If layer ℓ does not contextualize the representations at all, then $SelfSim_\ell(w) = 1$ (i.e., the representations are identical across all contexts). The more contextualized the representations are for w, the lower we would expect its self-similarity to be.

Definition 2 Let s be a sentence that is a sequence $\langle w_1, ..., w_n \rangle$ of n words. Let $f_\ell(s, i)$ be a function that maps $s[i]$ to its representation in layer ℓ of model f. The *intra-sentence similarity* of s in layer ℓ is

$$IntraSim_\ell(s) = \frac{1}{n} \sum_i \cos(\vec{s}_\ell, f_\ell(s, i))$$
$$\text{where } \vec{s}_\ell = \frac{1}{n} \sum_i f_\ell(s, i) \tag{2}$$

Put more simply, the *intra-sentence similarity* of a sentence is the average cosine similarity between its word representations and the sentence vector, which is just the mean of those word vectors. This measure captures how context-specificity manifests in the vector space. For example, if both $IntraSim_\ell(s)$ and $SelfSim_\ell(w)$ are low $\forall\, w \in s$, then the model contextualizes words in that layer by giving each one a context-specific representation that is still distinct from all other word representations in the sentence. If $IntraSim_\ell(s)$ is high but $SelfSim_\ell(w)$ is low, this suggests a less nuanced contextualization, where words in a sentence are contextualized simply by making their representations converge in vector space.

Definition 3 Let w be a word that appears in sentences $\{s_1, ..., s_n\}$ at indices $\{i_1, ..., i_n\}$ respectively, such that $w = s_1[i_1] = ... = s_n[i_n]$. Let $f_\ell(s, i)$ be a function that maps $s[i]$ to its representation in layer ℓ of model f. Where $[f_\ell(s_1, i_1)...f_\ell(s_n, i_n)]$ is the *occurrence matrix* of w and $\sigma_1...\sigma_m$ are the

first m singular values of this matrix, the *maximum explainable variance* is

$$MEV_\ell(w) = \frac{\sigma_1^2}{\sum_i \sigma_i^2} \qquad (3)$$

$MEV_\ell(w)$ is the proportion of variance in w's contextualized representations for a given layer that can be explained by their first principal component. It gives us an upper bound on how well a static embedding could replace a word's contextualized representations. The closer $MEV_\ell(w)$ is to 0, the poorer a replacement a static embedding would be; if $MEV_\ell(w) = 1$, then a static embedding would be a perfect replacement for the contextualized representations.

3.4 Adjusting for Anisotropy

It is important to consider isotropy (or the lack thereof) when discussing contextuality. For example, if word vectors were perfectly isotropic (i.e., directionally uniform), then $SelfSim_\ell(w) = 0.95$ would suggest that w's representations were poorly contextualized. However, consider the scenario where word vectors are so anisotropic that any two words have on average a cosine similarity of 0.99. Then $SelfSim_\ell(w) = 0.95$ would actually suggest the opposite – that w's representations were well contextualized. This is because representations of w in different contexts would on average be more dissimilar to each other than two randomly chosen words.

To adjust for the effect of anisotropy, we use three *anisotropic baselines*, one for each of our contextuality measures. For self-similarity and intra-sentence similarity, the baseline is the average cosine similarity between the representations of uniformly randomly sampled words from different contexts. The more anisotropic the word representations are in a given layer, the closer this baseline is to 1. For maximum explainable variance (MEV), the baseline is the proportion of variance in uniformly randomly sampled word representations that is explained by their first principal component. The more anisotropic the representations in a given layer, the closer this baseline is to 1: even for a random assortment of words, the principal component would be able to explain a large proportion of the variance.

Since contextuality measures are calculated for each layer of a contextualizing model, we calculate separate baselines for each layer as well.

We then subtract from each measure its respective baseline to get the *anisotropy-adjusted contextuality measure*. For example, the anisotropy-adjusted self-similarity is

$$Baseline(f_\ell) = \mathbb{E}_{x,y \sim U(\mathcal{O})}\left[\cos(f_\ell(x), f_\ell(y))\right]$$
$$SelfSim_\ell^*(w) = SelfSim_\ell(w) - Baseline(f_\ell) \qquad (4)$$

where $\mathcal{O}$ is the set of all word occurrences and $f_\ell(\cdot)$ maps a word occurrence to its representation in layer ℓ of model f. Unless otherwise stated, references to contextuality measures in the rest of the paper refer to the anisotropy-adjusted measures, where both the raw measure and baseline are estimated with 1K uniformly randomly sampled word representations.

4 Findings

4.1 (An)Isotropy

Contextualized representations are anisotropic in all non-input layers. If word representations from a particular layer were isotropic (i.e., directionally uniform), then the average cosine similarity between uniformly randomly sampled words would be 0 (Arora et al., 2017). The closer this average is to 1, the more anisotropic the representations. The geometric interpretation of anisotropy is that the word representations all occupy a narrow cone in the vector space rather than being uniform in all directions; the greater the anisotropy, the narrower this cone (Mimno and Thompson, 2017). As seen in Figure 1, this implies that in almost all layers of BERT, ELMo and GPT-2, the representations of all words occupy a narrow cone in the vector space. The only exception is ELMo's input layer, which produces static character-level embeddings without using contextual or even positional information (Peters et al., 2018). It should be noted that not all static embeddings are necessarily isotropic, however; Mimno and Thompson (2017) found that skipgram embeddings, which are also static, are not isotropic.

Contextualized representations are generally more anisotropic in higher layers. As seen in Figure 1, for GPT-2, the average cosine similarity between uniformly randomly words is roughly 0.6 in layers 2 through 8 but increases exponentially from layers 8 through 12. In fact, word representations in GPT-2's last layer are so anisotropic that any two words have on average an almost perfect cosine similarity! This pattern holds for BERT and
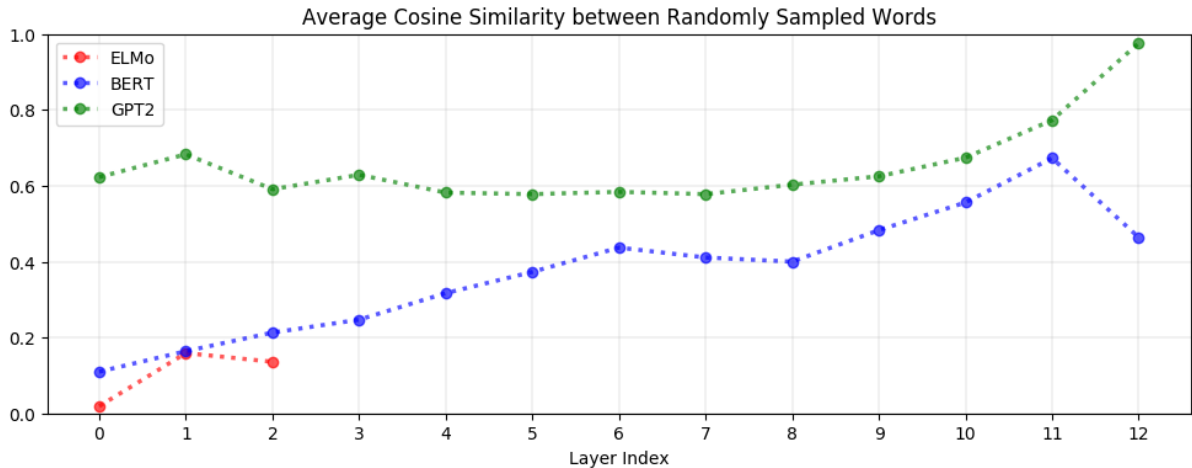
Figure 1: In almost all layers of BERT, ELMo, and GPT-2, the word representations are anisotropic (i.e., not directionally uniform): the average cosine similarity between uniformly randomly sampled words is non-zero. The one exception is ELMo's input layer; this is not surprising given that it generates character-level embeddings without using context. Representations in higher layers are generally more anisotropic than those in lower ones.

ELMo as well, though there are exceptions: for example, the anisotropy in BERT's penultimate layer is much higher than in its final layer.

Isotropy has both theoretical and empirical benefits for static word embeddings. In theory, it allows for stronger "self-normalization" during training (Arora et al., 2017), and in practice, subtracting the mean vector from static embeddings leads to improvements on several downstream NLP tasks (Mu et al., 2018). Thus the extreme degree of anisotropy seen in contextualized word representations – particularly in higher layers – is surprising. As seen in Figure 1, for all three models, the contextualized hidden layer representations are almost all more anisotropic than the input layer representations, which do not incorporate context. This suggests that high anisotropy is inherent to, or least a by-product of, the process of contextualization.

4.2 Context-Specificity

Contextualized word representations are more context-specific in higher layers. Recall from Definition 1 that the self-similarity of a word, in a given layer of a given model, is the average cosine similarity between its representations in different contexts, adjusted for anisotropy. If the self-similarity is 1, then the representations are not context-specific at all; if the self-similarity is 0, that the representations are maximally context-specific. In Figure 2, we plot the average self-similarity of uniformly randomly sampled words

in each layer of BERT, ELMo, and GPT-2. For example, the self-similarity is 1.0 in ELMo's input layer because representations in that layer are static character-level embeddings.

In all three models, the higher the layer, the lower the self-similarity is on average. In other words, the higher the layer, the more context-specific the contextualized representations. This finding makes intuitive sense. In image classification models, lower layers recognize more generic features such as edges while upper layers recognize more class-specific features (Yosinski et al., 2014). Similarly, upper layers of LSTMs trained on NLP tasks learn more task-specific representations (Liu et al., 2019a). Therefore, it follows that upper layers of neural language models learn more context-specific representations, so as to predict the next word for a given context more accurately. Of all three models, representations in GPT-2 are the most context-specific, with those in GPT-2's last layer being almost maximally context-specific.

Stopwords (e.g., 'the', 'of', 'to') have among the most context-specific representations. Across all layers, stopwords have among the lowest self-similarity of all words, implying that their contextualized representations are among the most context-specific. For example, the words with the lowest average self-similarity across ELMo's layers are 'and', 'of', ''s', 'the', and 'to'. This is relatively surprising, given that these words are not polysemous. This finding suggests that the variety
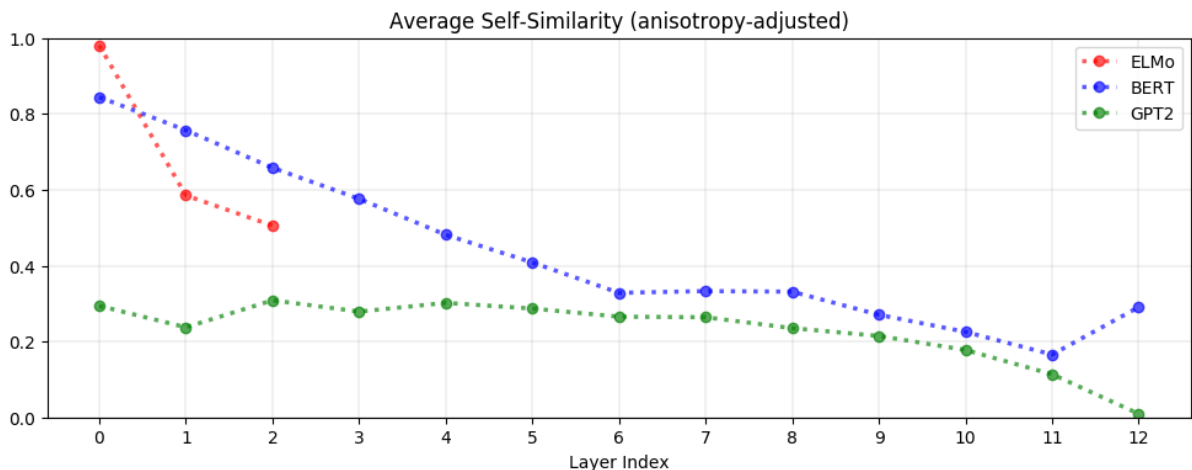
Figure 2: The average cosine similarity between representations of the same word in different contexts is called the word's *self-similarity* (see Definition 1). Above, we plot the average self-similarity of uniformly randomly sampled words after adjusting for anisotropy (see section 3.4). In all three models, the higher the layer, the lower the self-similarity, suggesting that contextualized word representations are more context-specific in higher layers.

of contexts a word appears in, rather than its inherent polysemy, is what drives variation in its contextualized representations. This answers one of the questions we posed in the introduction: ELMo, BERT, and GPT-2 are not simply assigning one of a finite number of word-sense representations to each word; otherwise, there would not be so much variation in the representations of words with so few word senses.

Context-specificity manifests very differently in ELMo, BERT, and GPT-2. As noted earlier, contextualized representations are more context-specific in upper layers of ELMo, BERT, and GPT-2. However, how does this increased context-specificity manifest in the vector space? Do word representations in the same sentence converge to a single point, or do they remain distinct from one another while still being distinct from their representations in other contexts? To answer this question, we can measure a sentence's intra-sentence similarity. Recall from Definition 2 that the intra-sentence similarity of a sentence, in a given layer of a given model, is the average cosine similarity between each of its word representations and their mean, adjusted for anisotropy. In Figure 3, we plot the average intra-sentence similarity of 500 uniformly randomly sampled sentences.

In ELMo, words in the same sentence are more similar to one another in upper layers. As word representations in a sentence become more context-specific in upper layers, the intra-sentence

similarity also rises. This suggests that, in practice, ELMo ends up extending the intuition behind Firth's (1957) distributional hypothesis to the sentence level: that because words in the same sentence share the same context, their contextualized representations should also be similar.

In BERT, words in the same sentence are more dissimilar to one another in upper layers. As word representations in a sentence become more context-specific in upper layers, they drift away from one another, although there are exceptions (see layer 12 in Figure 3). However, in all layers, the average similarity between words in the same sentence is still greater than the average similarity between randomly chosen words (i.e., the anisotropy baseline). This suggests a more nuanced contextualization than in ELMo, with BERT recognizing that although the surrounding sentence informs a word's meaning, two words in the same sentence do not necessarily have a similar meaning because they share the same context.

In GPT-2, word representations in the same sentence are no more similar to each other than randomly sampled words. On average, the unadjusted intra-sentence similarity is roughly the same as the anisotropic baseline, so as seen in Figure 3, the anisotropy-adjusted intra-sentence similarity is close to 0 in most layers of GPT-2. In fact, the intra-sentence similarity is highest in the input layer, which does not contextualize words at all. This is in contrast to ELMo and BERT, where the
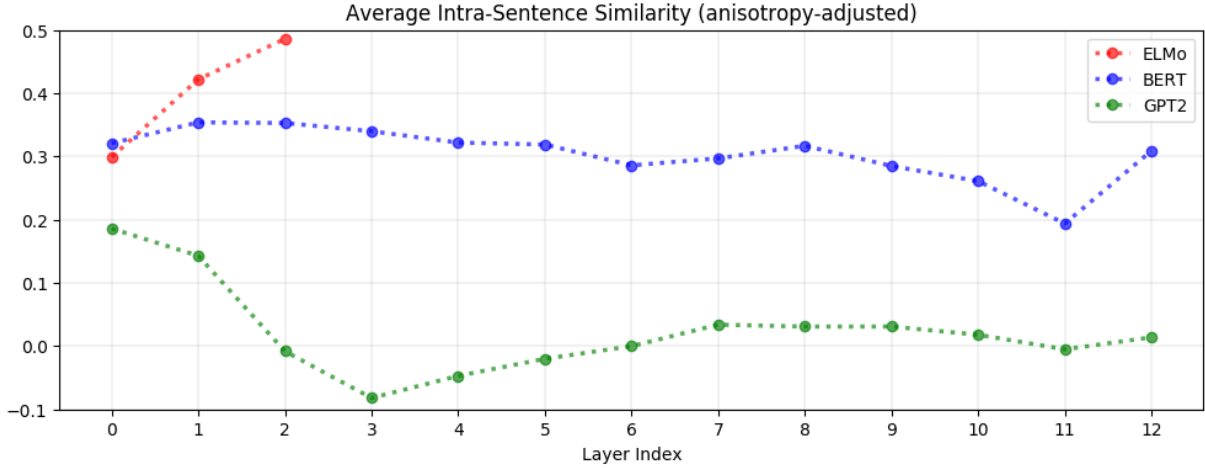
Figure 3: The *intra-sentence similarity* is the average cosine similarity between each word representation in a sentence and their mean (see Definition 2). Above, we plot the average intra-sentence similarity of uniformly randomly sampled sentences, adjusted for anisotropy. This statistic reflects how context-specificity manifests in the representation space, and as seen above, it manifests very differently for ELMo, BERT, and GPT-2.

average intra-sentence similarity is above 0.20 for all but one layer.

As noted earlier when discussing BERT, this behavior still makes intuitive sense: two words in the same sentence do not necessarily have a similar meaning simply because they share the same context. The success of GPT-2 suggests that unlike anisotropy, which accompanies context-specificity in all three models, a high intra-sentence similarity is not inherent to contextualization. Words in the same sentence can have highly contextualized representations without those representations being any more similar to each other than two random word representations. It is unclear, however, whether these differences in intra-sentence similarity can be traced back to differences in model architecture; we leave this question as future work.

4.3 Static vs. Contextualized

On average, less than 5% of the variance in a word's contextualized representations can be explained by a static embedding. Recall from Definition 3 that the *maximum explainable variance* (MEV) of a word, for a given layer of a given model, is the proportion of variance in its contextualized representations that can be explained by their first principal component. This gives us an upper bound on how well a static embedding could replace a word's contextualized representations. Because contextualized representations are anisotropic (see section 4.1), much of the variation *across all words* can be explained by a sin-

gle vector. We adjust for anisotropy by calculating the proportion of variance explained by the first principal component of uniformly randomly sampled word representations and subtracting this proportion from the raw MEV. In Figure 4, we plot the average anisotropy-adjusted MEV across uniformly randomly sampled words.

In no layer of ELMo, BERT, or GPT-2 can more than 5% of the variance in a word's contextualized representations be explained by a static embedding, on average. Though not visible in Figure 4, the raw MEV of many words is actually below the anisotropy baseline: i.e., a greater proportion of the variance across all words can be explained by a single vector than can the variance across all representations of a single word. Note that the 5% threshold represents the best-case scenario, and there is no theoretical guarantee that a word vector obtained using GloVe, for example, would be similar to the static embedding that maximizes MEV. This suggests that contextualizing models are not simply assigning one of a finite number of word-sense representations to each word – otherwise, the proportion of variance explained would be much higher. Even the average raw MEV is below 5% for all layers of ELMo and BERT; only for GPT-2 is the raw MEV non-negligible, being around 30% on average for layers 2 to 11 due to extremely high anisotropy.

Principal components of contextualized representations in lower layers outperform GloVe and FastText on many benchmarks. As noted
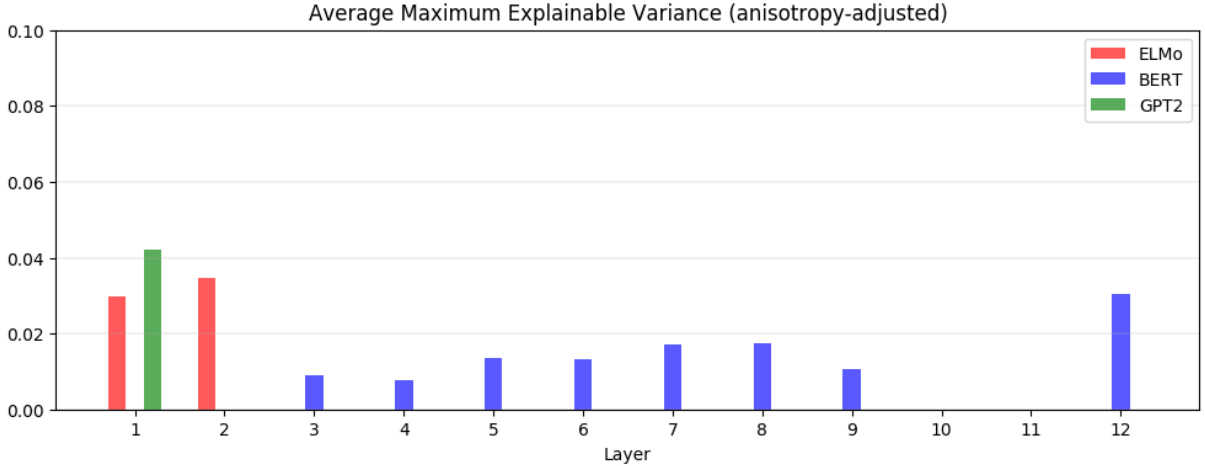
Figure 4: The *maximum explainable variance* (MEV) of a word is the proportion of variance in its contextualized representations that can be explained by their first principal component (see Definition 3). Above, we plot the average MEV of uniformly randomly sampled words after adjusting for anisotropy. In no layer of any model can more than 5% of the variance in a word's contextualized representations be explained by a static embedding.

Static Embedding	SimLex999	MEN	WS353	RW	Google	MSR	SemEval2012(2)	BLESS	AP
GloVe	0.194	0.216	0.339	0.127	0.189	0.312	0.097	0.390	0.308
FastText	0.239	**0.239**	**0.432**	0.176	0.203	0.289	0.104	0.375	0.291
ELMo, Layer 1	0.276	0.167	0.317	0.148	0.170	0.326	0.114	**0.410**	0.308
ELMo, Layer 2	0.215	0.151	0.272	0.133	0.130	0.268	0.132	0.395	0.318
BERT, Layer 1	0.315	0.200	0.394	**0.208**	**0.236**	**0.389**	**0.166**	0.365	**0.321**
BERT, Layer 2	**0.320**	0.166	0.383	0.188	0.230	0.385	0.149	0.365	**0.321**
BERT, Layer 11	0.221	0.076	0.319	0.135	0.175	0.290	0.149	0.370	0.289
BERT, Layer 12	0.233	0.082	0.325	0.144	0.184	0.307	0.144	0.360	0.294
GPT-2, Layer 1	0.174	0.012	0.176	0.183	0.052	0.081	0.033	0.220	0.184
GPT-2, Layer 2	0.135	0.036	0.171	0.180	0.045	0.062	0.021	0.245	0.184
GPT-2, Layer 11	0.126	0.034	0.165	0.182	0.031	0.038	0.045	0.270	0.189
GPT-2, Layer 12	0.140	-0.009	0.113	0.163	0.020	0.021	0.014	0.225	0.172

Table 1: The performance of various static embeddings on word embedding benchmark tasks. The best result for each task is in bold. For the contextualizing models (ELMo, BERT, GPT-2), we use the first principal component of a word's contextualized representations in a given layer as its static embedding. The static embeddings created using ELMo and BERT's contextualized representations often outperform GloVe and FastText vectors.

earlier, we can create static embeddings for each word by taking the first principal component (PC) of its contextualized representations in a given layer. In Table 1, we plot the performance of these *PC static embeddings* on several benchmark tasks[2]. These tasks cover semantic similarity, analogy solving, and concept categorization: SimLex999 (Hill et al., 2015), MEN (Bruni et al., 2014), WS353 (Finkelstein et al., 2002), RW (Luong et al., 2013), SemEval-2012 (Jurgens et al., 2012), Google analogy solving (Mikolov et al., 2013a) MSR analogy solving (Mikolov et al., 2013b), BLESS (Baroni and Lenci, 2011) and AP (Almuhareb and Poesio, 2004). We leave out layers 3 - 10 in Table 1 because their performance is

between those of Layers 2 and 11.

The best-performing PC static embeddings belong to the first layer of BERT, although those from the other layers of BERT and ELMo also outperform GloVe and FastText on most benchmarks. For all three contextualizing models, PC static embeddings created from lower layers are more effective those created from upper layers. Those created using GPT-2 also perform markedly worse than their counterparts from ELMo and BERT. Given that upper layers are much more context-specific than lower layers, and given that GPT-2's representations are more context-specific than ELMo and BERT's (see Figure 2), this suggests that the PCs of highly context-specific representations are less effective on traditional benchmarks. Those derived from less context-specific represen-

---
[2]The Word Embeddings Benchmarks package was used for evaluation.

tations, such as those from Layer 1 of BERT, are much more effective.

5 Future Work

Our findings offer some new directions for future work. For one, as noted earlier in the paper, Mu et al. (2018) found that making static embeddings more isotropic – by subtracting their mean from each embedding – leads to surprisingly large improvements in performance on downstream tasks. Given that isotropy has benefits for static embeddings, it may also have benefits for contextualized word representations, although the latter have already yielded significant improvements despite being highly anisotropic. Therefore, adding an anisotropy penalty to the language modelling objective – to encourage the contextualized representations to be more isotropic – may yield even better results.

Another direction for future work is generating static word representations from contextualized ones. While the latter offer superior performance, there are often challenges to deploying large models such as BERT in production, both with respect to memory and run-time. In contrast, static representations are much easier to deploy. Our work in section 4.3 suggests that not only it is possible to extract static representations from contextualizing models, but that these extracted vectors often perform much better on a diverse array of tasks compared to traditional static embeddings such as GloVe and FastText. This may be a means of extracting some use from contextualizing models without incurring the full cost of using them in production.

6 Conclusion

In this paper, we investigated how contextual contextualized word representations truly are. For one, we found that upper layers of ELMo, BERT, and GPT-2 produce more context-specific representations than lower layers. This increased context-specificity is always accompanied by increased anisotropy. However, context-specificity also manifests differently across the three models; the anisotropy-adjusted similarity between words in the same sentence is highest in ELMo but almost non-existent in GPT-2. We ultimately found that after adjusting for anisotropy, on average, less than 5% of the variance in a word's contextualized representations could be explained by a static

embedding. This means that even in the best-case scenario, in all layers of all models, static word embeddings would be a poor replacement for contextualized ones. These insights help explain some of the remarkable success that contextualized representations have had on a diverse array of NLP tasks.

Acknowledgments

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings SemEval@ NAACL-HLT*. pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings SemEval@ COLING*. pages 81–91.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. Sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 385–393.

Abdulrahman Almuhareb and Massimo Poesio. 2004. Attribute-based and value-based clustering: An evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics, pages 1–10.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research* 49:1–47.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems* 20(1):116–131.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis* .

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* 41(4):665–695.

David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 356–364.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*. pages 171–180.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*. pages 2177–2185.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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. *arXiv preprint arXiv:1907.11692* .

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. pages 104–113.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. 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.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 2873–2878.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners .

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJzSgnRcKX.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* .

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*. pages 3320–3328.