Iran University of Science and Technology
Deep Learning (Spring 2020)

Isotropic Contextual Word Representation in BERT

Sara Rajaee
Department of Computer Engineering
Iran University of Science
and Technology
sara_rajaee@comp.iust.ac.ir

Marzieh Sheikhi
Department of Computer Engineering
Iran University of Science
and Technology
marzieh_sheikhi@comp.iust.ac.ir

Abstract

Recent advance in representation learning shows that isotropic (i.e., unit-variance and uncorrelated) embeddings can significantly improve performance on downstream tasks with faster convergence and better generalization. In this project, we proposed an algorithm to apply in both the pre-trained and fine-tuning phase. Our algorithm attempts to make the representations isotropic. We analyze the isotropy of BERT model in both the pre-trained and fine-tune phases. Our result demonstrates that our algorithm can improve the pre-trained representations isotropy. But in the fine-tuning phase, we can not enhance the performance of the model.

1 Introduction

BERT [1] has an important effect on NLP tasks. This model is pre-trained on vast amounts of unannotated data and fine-tuned on much smaller annotated datasets. During pre-training on large-scale corpora, it learns to generate powerful internal representations, including fine-grained contextual word embedding. In this project, we want to analyze the isotropy of contextual word embeddings in pre-trained and fine-tuned BERT. This property improves the convergence rate of the optimizer algorithm and makes the model interpretation easier. Therefore, we attempt to propose an algorithm to enhance the performance of word representations of the model in the pre-trained and fine-tuning phase.

2 Related work/Background

To explain our proposed method, we express some background concepts. We will then describe recent works that had proposed an approach to overcome or solve the problem.

2.1 Background

2.1.1 BERT model

In this project, we used the BERT model to analyze the contextual representations. BERT, or Bidirectional Encoder Representations from Transformers, is essentially a new method of training language models. The contextual word representation depends on the context where the word occurs, which leads the same word in different contexts can have different representations.

BERT architecture builds on top of the Transformer. We currently have two variants available:

(LateX template borrowed from NeurIPS 2019)

- BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
- BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters

We should mention that the fine-tuning approach isn't the only way to use BERT. We can use the pre-trained BERT to extract features of each word of sentences. In this project, we use both the pre-trained and fine-tuned BERT model.

2.1.2 Isotropy

We studied a feature of vector space that is called Isotropy. A vector space is isotropic if it is uniformly distributed in the space. According to [2], the idea of isotropy comes from the partition function defined in [5],

$$Z(c) = \sum_{w \in \mathcal{V}} \exp\left(c^{\top} v(w)\right),$$

where $Z(\,c)$ should approximately be a constant with any unit vector c and $\{v(\,w)\,, w \in V\}$ is the set of word representations. Hence, as defined in [2], the isotropy is measured as follows,

$$I(\{v(w)\}) = \frac{\min_{\|c\|=1} Z(c)}{\max_{\|c\|=1} Z(c)},$$

where $I(\,\{v(\,w)\,\})$ ranges from 0 to 1, and $I(\,\{v(\,w)\,\})$ closer to 1 indicates that $\{v(\,w)\,\}$ is more isotropic. The intuition behind our postprocessing algorithm can also be motivated by letting $I(\,\{v(\,w)\,\}) \longrightarrow 1$. It would be better to mention that, we approximated the set C as the set of eigenvectors of $V^T V$, where V is word representations' vector space. Also, we understood that if a word representations space is isotropic, it can help gradient descent based optimizer to converge a better result and improves the model training speed.

2.2 Recent works

The main study in the improvement of isotropy is [2], which was used on static word embeddings. In [2], they made zero-mean data. Next, they eliminated the effects of the dominant directions to project the data into weak directions. The authors in [3] proposed an approach that used the algorithm in [2]. They applied the algorithm [2] twice with data transformation between them by using PCA. Another study, which is the only approach on contextual word embeddings, was proposed in [4]. They assumed that the absolute correlation coefficient matrix is a block-diagonal binary matrix. They applied batch normalization on each block and transformed the data. Their proposed approach is called Isotropic Batch normalization (IsoBN).

3 Proposed method

In this section, we describe our proposed post-process algorithm on the pre-trained BERT model. We will then express how we use our method in the fine-tuning phase of the BERT model.

Post-processing and dimensionality reduction techniques in word embeddings have primarily been based on the principal component analysis (PCA). More specifically, we explain the Post-process algorithm in [2] on static word embeddings. Then, we proposed our algorithm along our motivations to evaluate it on a contextual-based model.

To improve the isotropy of the static word embeddings, the authors of [2] proposed an algorithm that is represented in Algorithm 1. In this algorithm, the mean of word vectors was subtracted from each of them to make all vectors zero-mean. Next, they applied the PCA algorithm on the zero-mean word vectors to find the most dominant directions, which exert a strong influence on the other vectors in the same way. They subtracted the effects of the dominant direction from zero-mean word vectors. i.e., they projected word embeddings toward the weak directions rather than the dominant directions. This algorithm improves the static word embeddings' isotropy. We evaluate this algorithm on the output of the Pre-trained BERT model. The result demonstrates that this approach was not suitable

for contextual word representations. This may cause of that in static word embeddings, the dominated dimensions encoded word frequency.

Algorithm 1: Post-processing algorithm in [1]

Input: Word representations $\{v(w), w \in \mathcal{V}\}$, a threshold parameter D

Compute the mean of the word representations as μ

Subtract the mean from word embedding as $\{\tilde{v}(w)|w \in \mathcal{V},\ \tilde{v}(w) = v(w) - \mu\}$,

Compute the PCA components: $u_1, \dots, u_d \leftarrow PCA(\{\tilde{v}(w), w \in \mathcal{V}\})$

Preprocess on representations: $v'(w) \leftarrow \tilde{v}(w) - \sum_{i=1}^{D}(u_i^T v(w))u_i$

Output: Processed representations $v'(w)$

To renovate this algorithm to make it compatible with the contextual word representations, we use clustering algorithms to cluster similar representations together and apply the same process on similar embeddings. For each cluster, we subtract the mean of clustered word embeddings from their elements as $\tilde{v}(w)$ to make zero-mean word representations for each one. After that, we apply the PCA algorithm and select the first D PCA's components. We eliminate the most D dominated directions of the $\tilde{v}(w)$. We use mean because it has effects on the word similarities. Also, we exert the PCA algorithm to get the most dominant components with respect to the direction. We put $D = d/100$ that d is number of word representation features. Algorithm 2 shows the pseudo-code of our proposed algorithm.

Algorithm 2: Post-processing algorithm on contextual word embeddings

Input: Word representations $\{v(w), w \in \mathcal{V}\}$, a threshold parameter D, n number of clusters

Cluster the similar word representations

Compute the mean of each cluster word representations as $\mu = (\mu_1, \dots \mu_n)$

Subtract the mean of each cluster from their word embedding as $\{\tilde{v}(w), w \in \mathcal{V}\}$,

Compute the PCA components: $u_1, \dots, u_d \leftarrow PCA(\{\tilde{v}(w), w \in \mathcal{V}\})$

Preprocess on representations: $v'(w) \leftarrow \tilde{v}(w) - \sum_{i=1}^{D}(u_i^T v(w))u_i$

Output: Processed representations $v'(w)$

To fine-tune BERT model, we add our proposed algorithm between the model and the classification layer same as Figure 1. We use our algorithm on the fine-tuning phase by applying the algorithm on each batch of data. Also, we should mention that our algorithm is executed on the CLS token of sentences.

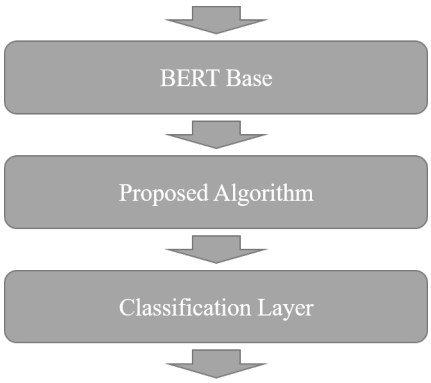


Figure 1: Isotropy of Pre-trained BERT Base layers

4 Results

In the following section, we express tasks and datasets we used to evaluate our approach. Then, we discuss about our experiments and their results.

4.1 Dataset

We used different datasets to evaluate our proposed method, that the results are illustrated in the next section. We examine the isotropy of BERT model on STS, SST-2, WiC, RTE, and SICK. More specifically, we use the SemEval 2015 and STS-Benchmark to investigate the performance of different post-processing modes on BERT model.

4.2 Experiments

4.2.1 Feature Extraction

We measured the isotropy of the contextual word representations space of each layer of the pre-trained BERT model. As can be seen in Figure 2, the word representations of each layer are extremely anisotropic. 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 [6].

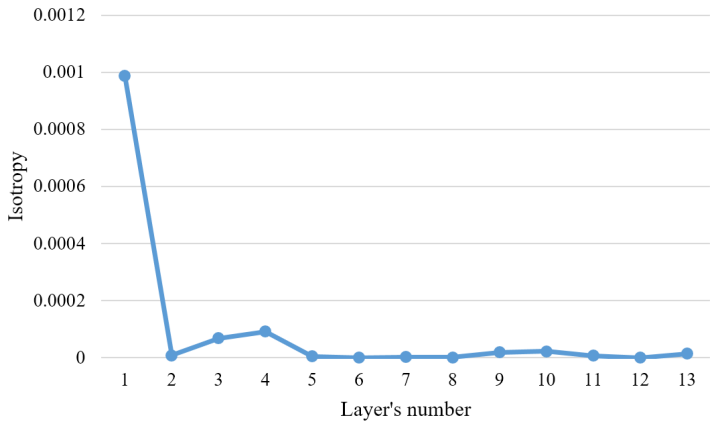


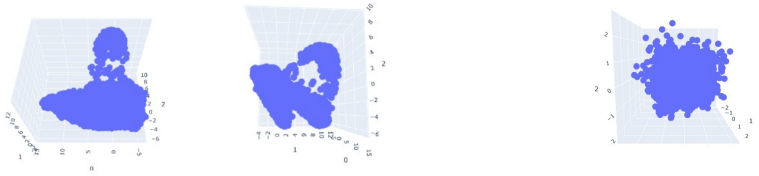
Figure 2: Isotropy of Pre-trained BERT Base layers

We implemented K-means and GMM (Gaussian Mixture Modelling) as the clustering algorithms. The below table illustrates the results of the implementation of our proposed algorithm in a different mode. It is better to mention that we fine-tuned the number of clusters in the K-means algorithm to get a better result.

	SEMEVAL 2015 –TASK 2			STS-B		
	Pearson	Spearman	Isotropy	Pearson	Spearman	Isotropy
Pre-trained BERT	56.7	53.68	1.35 e-5	56.16	54.09	3.36 e-5
Mean	60.51	56.81	4.60e-6	60.08	57.00	2.16 e-6
Mean + PCA	55.81	53.23	7.30 e-8	47.40	46.67	9.61 e-13
K-means + Mean	66.58	62.38	0.33	68.43	64.25	0.2376
GMM + Mean + PCA	69.18	65.51	0.84	70.53	67.32	**0.6380**
K-means + Mean + PCA	**69.84**	**66.33**	**0.85**	**70.75**	**67.50**	0.6353

As can be seen in the below table, the Pearson and Spearman Correlations and isotropy are improved by using our proposed algorithm. Also, it demonstrates that using the K-means algorithm causes better performance than the GMM algorithm. Another point is that algorithms which are used for static word embeddings are not suitable for contextual word representations.

Figure 3 demonstrates the distribution of the representation before and after applying our proposed method with the K-means clustering algorithm. As can be seen, the shape of the representations is approximately rounded.



(a) Before applying (b) After applying

Figure 3: Improvement of isotropy by applying proposed algorithm

The below table illustrates the performance of our proposed algorithm on the Semantic Similarity task in comparison to the pre-trained BERT model.

	Pre-trained BERT Base			Proposed Algorithm		
Dataset	Pearson	Spearman	Isotropy	Pearson	Spearman	Isotropy
STS2012	45.46	43.53	2.97 e-5	**72.67**	**64.73**	**0.55**
STS2013	62.39	59.50	2.60 e-4	**69.88**	**68.89**	**0.46**
STS2014	56.71	53.36	3.84 e-6	**66.11**	**62.14**	**0.53**
STS2015	56.70	53.68	1.35 e-5	**69.84**	**66.33**	**0.85**
STS2016	61.33	61.11	1.01 e-4	**67.46**	**66.68**	**0.48**
STS-Benchmark	56.16	54.09	3.36 e-5	**70.75**	**67.50**	**0.63**
SICK	62.32	59.38	2.93 e-4	**66.47**	**63.07**	**0.49**

Also, for classification tasks, we evaluate our proposed algorithm on SST-2 task. The below table, represents our implementation results.

	Pre-trained BERT Base		Proposed Algorithm	
	Accuracy	Isotropy	Accuracy	Isotropy
SST-2	55.67	7.97 e-6	**56.08**	**84.36**

4.2.2 Fine tuning

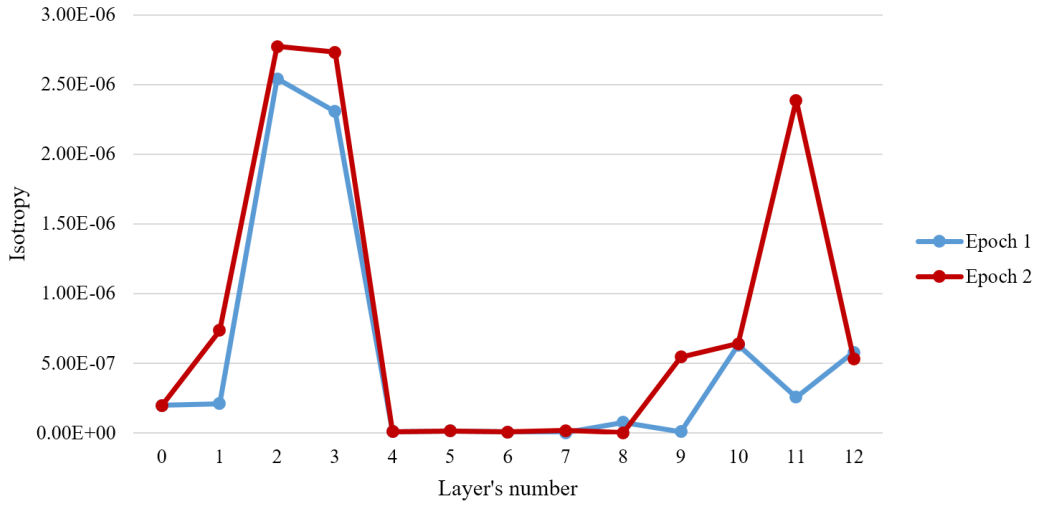


Figure 4: Isotropy of Fine-tuned BERT Base layers

In the fine-tuning phase, we examine the isotropy of BERT layers on the STS-Benchmark dataset in two epochs. Figure 4 illustrates the results of the each epoch. The results represent that the BERT is an anisotropic model, even fine-tuned by a task.

The below table illustrates our result on RTE and WiC tasks.

	BERT Base		Proposed Algorithm	
	Accuracy	Isotropy	Accuracy	Isotropy
RTE (Re-Imp)	64.70	4.86 e-5	62.80	**0.29**
WiC (Re-Imp)	64.04	8.62 e-4	62.80	**0.13**

5 Discussion

In this project, we proposed an algorithm that uses a clustering algorithm to cluster the word representations. In each cluster, we make data zero-mean and subtract the dominant directions to make them round. In comparison to other studies, our proposed method can apply on contextual representations and BERT model. Also, we are the first approach that can improve the performance of the model on contextual representations.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805, 2018*.

[2] J. Mu, S. Bhat, and P. Viswanath. All-but-the-top: simple and effective post-processing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

[3] Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. *Proceedings of the 4th Workshop on Representation Learning for NLP*.

[4] Wenxuan Zhou, Bill Yuchen Lin, Xiang Ren. IsoBN: Fine-Tuning BERT with Isotropic Batch Normalization. *arXiv preprint arXiv:2005.02178*, 2020.

[5] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings.*Transactions of the Association for Computational Linguistics*, 4:385–399, 2016. ISSN 2307-387X. URL: *https://transacl.org/ojs/ index.php/tacl/article/view/742*.

[6] Ethayarajh, Kawin. How Contextual Are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings.*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.*