

# On Dimensional Linguistic Properties of the Word Embedding Space

**Vikas Raunak\***

Carnegie Mellon University  
vraunak@cs.cmu.edu

**Vaibhav Kumar\***

Carnegie Mellon University  
vaibhav2@cs.cmu.edu

**Vivek Gupta**

University of Utah  
vgupta@cs.utah.edu

**Florian Metze**

Carnegie Mellon University  
fmetze@cs.cmu.edu

## Abstract

Word embeddings have become a staple of several natural language processing tasks, yet much remains to be understood about their properties. In this work, we analyze word embeddings in terms of their principal components and arrive at a number of novel conclusions. In particular, we characterize the utility of variance explained by the principal components (widely used as a fundamental tool to assess the quality of the resulting representations) as a proxy for downstream performance. Further, through dimensional linguistic probing of the embedding space, we show that the syntactic information captured by a principal component does not depend on the amount of variance it explains. Consequently, we investigate the limitations of variance based embedding post-processing techniques and demonstrate that such post-processing is counter-productive in a number of scenarios such as sentence classification and machine translation tasks. Finally, we offer a few guidelines on variance based embedding post-processing. We have released the source code along with the paper<sup>1</sup>.

## 1 Introduction

Word embeddings have revolutionized natural language processing by representing words as dense real-valued vectors in a low dimensional space. Pre-trained word embeddings such as Glove (Pennington et al., 2014), word2vec (Mikolov et al., 2013) and fasttext (Bojanowski et al., 2017), trained on large corpora are readily available for use in a variety of tasks. Subsequently, there has been emphasis on post-processing the embeddings to improve their performance on downstream tasks (Mu and Viswanath, 2018) or to induce linguistic properties (Mrkšić et al.; Faruqui et al., 2015).

In particular, the Principal Component Analysis (PCA) based post-processing algorithm proposed by (Mu and Viswanath, 2018) has led to significant gains in word and sentence similarity tasks, and has also proved useful in dimensionality reduction (Raunak, 2017). Similarly, understanding the geometry of word embeddings is another area of active research (Mimno and Thompson, 2017). Researchers have tried to ascertain the importance of dimensionality for word embeddings, with results from (Yin and Shen, 2018) answering the question of optimal dimensionality selection. In contrast to previous work, we explore the dimensional properties of existing pre-trained word embeddings through their principal components. Specifically, our contributions are as follows:

1. We analyze the word embeddings in terms of their principal components and demonstrate that their performance on both word similarity and sentence classification tasks saturates well before the full dimensionality.
2. We demonstrate that the amount of variance captured by the principal components is a poor representative for the downstream performance of the embeddings constructed using the very same principal components.
3. We investigate the reasons behind the aforementioned result through syntactic information based dimensional linguistic probing tasks (Conneau et al., 2018) and demonstrate that the syntactic information captured by a principal component is independent of the amount of variance it explains.
4. We point out the limitations of applying variance based post-processing (Mu and Viswanath, 2018) and demonstrate that it leads to a decrease in performance in sentence classification and machine translation

\*equal contribution

<sup>1</sup> Code Link:

[https://github.com/aclsrw/anonymized\\_code](https://github.com/aclsrw/anonymized_code)

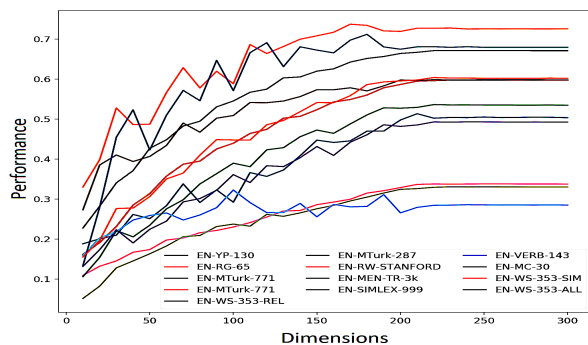


Figure 1:  $\rho$  on Word Similarity Tasks

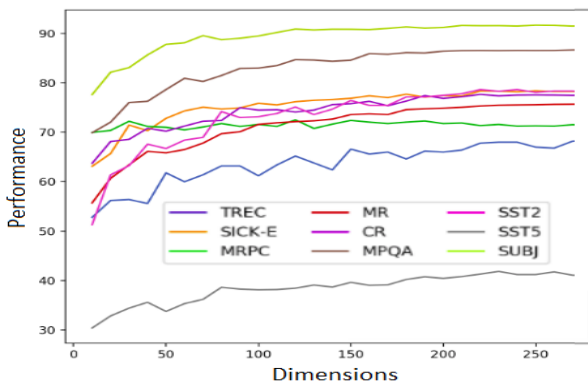


Figure 2: Accuracy on Sentence Classification Tasks

tasks, restricting its efficacy to only semantic similarity tasks.

In Section 1, we provide an introduction to the problem statement. In Section 2 we discuss dimensional properties of word embeddings. In Section 3 we conduct a variance based analysis by measuring performance of word embeddings on downstream tasks. In Section 4 we move on to dimensional linguistic probing tasks followed by Section 5 where we discuss the post-processing algorithm, and finally conclude in Section 6.

## 2 Dimensional Properties of the Word Embedding Space

Principal components provide a natural basis for studying the properties of an embedding space. In this work, we refer to the properties pertaining to the principal components of the embedding space as dimensional properties. We study such dimensional properties in a number of different contexts such as word similarity, sentence classification and machine translation tasks.

For experiments in this section, we use Glove embeddings (trained on Wikipedia 2014 + Gi-

gaword 5<sup>2</sup>). Subsequently, we also use fast-text (trained on Wikipedia, UBMC webbase corpus and statmt.org news dataset<sup>3</sup>) and Word2vec (trained on the GoogleNews dataset<sup>4</sup>) embeddings. Details of the corresponding datasets and evaluation tasks have been omitted due to space limit. Please refer to [Conneau and Kiela \(2018\)](#) for the details on sentence classification tasks and the classification algorithm and [Faruqui and Dyer \(2014\)](#) for word similarity benchmarks. Further, each of our experiments (except Machine Translation in Section 5.2) are deterministic in nature.

### 2.1 Word Similarity Tasks

Figure 1 shows the performance of word embeddings on 13 word similarity benchmarks ([Faruqui and Dyer, 2014](#)). The dimensions vary along the X-axis and each new evaluation cumulatively adds 10 more principal components to the embeddings (thus, there are 30 measurements for each dataset, ranging from word embeddings constructed using the first 10 principal components to full 300 principal components). The performance is measured using Spearman’s rank correlation coefficient ( $\rho$ ) between the human assigned and cosine similarity based rankings of the word vectors.

### 2.2 Sentence Classification Tasks

Figure 2 shows the performance (Test accuracy) on 9 standard downstream sentence classification tasks ([Conneau and Kiela, 2018](#)) using the same procedure for constructing word embeddings as done in 2.1. Further, sentence vectors were constructed using an average of the contained word embeddings, which has been demonstrated to be a very strong baseline for downstream tasks ([Arora et al., 2017](#)). From Figures 1, 2 it is evident that the performance saturates consistently at around 200 dimensions, after which adding new principal components does not lead to much gain in performance. This implies redundancy (not *noise*) among the dimensions. Further, this also suggests a simple strategy to reduce embedding size wherein one third of the components could be reliably removed without affecting the performance on word similarity or sentence classification tasks, leading to approximately 33% memory reduction.

<sup>2</sup> <https://stanford.io/2Gdv8uo>

<sup>3</sup> <https://bit.ly/2FMTB4N>

<sup>4</sup> <https://bit.ly/2esteWf>

Table 1: Test accuracy of embeddings composed of **Top-100 (T)**, **Middle-100 (M)** and **Bottom-100 (B)** principal components on sentence classification datasets. The values inside [] on the right side of each embedding type describes the variance explained by the included principal components. The highlighted cells correspond to one of the three cases - *M outperforms T* (orange), *B outperforms T* (red) and *B outperforms M* (yellow)

	MR	CR	SUBJ	MPQA	SST2	SST5	TREC	SICK-E	MRPC
<b>Glove-T</b> [0.529]	70.74	73.67	90.1	81.58	72.49	37.24	61.8	75.71	71.94
<b>Glove-M</b> [0.371]	<b>72.98</b>	<b>75.04</b>	87.76	<b>84.07</b>	<b>75.34</b>	<b>40.5</b>	57.6	<b>76.5</b>	71.42
<b>Glove-B</b> [0.100]	67.62	73.01	83.68	<b>81.61</b>	69.52	36.11	57.0	72.82	70.96
<b>Word2vec-T</b> [0.628]	74.34	76.29	89.88	85.07	77.16	40.36	70.0	75.46	71.48
<b>Word2vec-M</b> [0.221]	72.91	73.43	82.39	82.76	72.65	38.69	66.0	70.53	71.36
<b>Word2vec-B</b> [0.150]	71.42	<b>74.25</b>	<b>82.47</b>	81.05	73.48	38.46	<b>72.2</b>	<b>74.3</b>	71.01
<b>FastText-T</b> [0.745]	69.42	67.76	87.69	84.64	74.35	36.83	74.8	66.04	70.61
<b>FastText-M</b> [0.162]	68.88	65.3	81.74	81.45	72.1	35.57	65.2	65.01	68.29
<b>FastText-B</b> [0.093]	66.45	64.21	79.89	79.83	69.96	31.22	<b>69.4</b>	63.77	67.94

Table 2: Performance on sentence classification tasks of various embeddings and their Post Processed (PPA) counterparts. The red colored cells denote the cases where the original embeddings outperformed their Post Processed (PPA) counterparts.

	MR	CR	SUBJ	MPQA	SST2	SST5	TREC	SICK-E	MRPC
<b>Glove (300 dim)</b>	75.7	77.48	91.76	86.66	78.03	41.0	68.8	78.49	70.61
<b>PPA on Glove</b>	<b>75.57</b>	77.48	<b>91.01</b>	86.67	<b>77.98</b>	<b>40.72</b>	<b>65.8</b>	78.53	71.59
<b>Word2vec (300 dim)</b>	77.65	79.23	90.76	88.30	79.68	42.44	82.6	78.24	72.64
<b>PPA on Word2vec</b>	<b>77.33</b>	79.5	<b>90.59</b>	<b>88.12</b>	<b>79.41</b>	42.71	83.4	78.26	<b>72.58</b>
<b>Fasttext (300 dim)</b>	74.16	71.63	89.56	87.12	79.24	39.14	79.4	72.34	70.14
<b>PPA on Fasttext</b>	74.59	71.63	<b>89.4</b>	<b>86.9</b>	<b>79.13</b>	39.64	80.2	72.36	<b>70.09</b>

### 3 Variance Based Analysis

In this section, we characterize the redundancy observed in Section 2, in terms of variance of the principal components. Specifically, we measure downstream performance (on the sentence classification tasks of Section 2.2) of word embeddings against the amount of variance captured by the principal components. For each of the embedding types, we first construct word embeddings using only top 100 principal components (T), the middle 100 principal components (M) and the bottom 100 principal components (B). The three sets of embeddings differ significantly, across the embedding types, in terms of the variance explained by the principal components used in their construction. The results are presented in Table 1. The highlighted cells in Table 1 show that, in a number of cases, embeddings built using the middle (M) and the bottom 100 principal components (B) outperform the embeddings constructed using the top 100 principal components (T). Although, variance explained by the principal components is widely used as a fundamental tool to assess the quality of the corresponding representations (Jolliffe and

Cadima, 2016), these results demonstrate that for word embeddings, the variance explained by the principal components is a poor representative of downstream performance.

### 4 Dimensional Linguistic Probing Tasks

A hypothesis to explain the better performance of M and B embeddings (Table 1) in the earlier section is that the syntactic information required for downstream sentence classification tasks is distributed independently with respect to the principal components. To explore this hypothesis, we propose to leverage two classification based linguistic probing tasks, namely TreeDepth and TopConst (Conneau et al., 2018), which are designed to test whether sentence embeddings are sensitive to syntactic properties of the encoded sentences. The TreeDepth task tests whether the model can predict the depth of the hierarchical syntactic structure of the sentence, whereas in TopConst, a sentence must be classified in terms of the sequence of its constituents occurring immediately below the sentence node of its hierarchical structure. To evaluate the syntactic information

contained in each of the principal components, we first construct one-dimensional word embeddings by projecting word vectors onto a single principal component and then use sentence vectors constructed by using these embeddings for solving the TreeDepth and TopConst tasks. Figure 3 depicts the scores (Test accuracy) on TopConst and TreeDepth tasks respectively. Evidently, no single principal component (dimension) achieves a significantly higher score in any of the two tasks and the performance across the dimensions does not have any particular trend (increasing or decreasing). This validates the hypothesis that the principal components do not vary disproportionately in terms of the syntactic information contained.

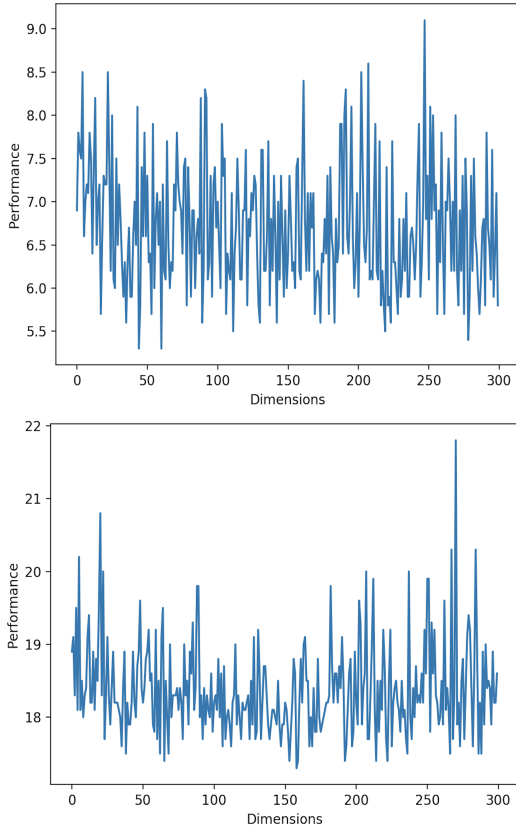


Figure 3: Analysis of individual principal components on the two syntactic information based linguistic probing tasks: TopConst(top) and TreeDepth(bottom). The Y-axis represents the Test accuracy on the two tasks.

## 5 The Post Processing Algorithm (PPA)

In this section, we first describe and then evaluate the post-processing algorithm (PPA) proposed in (Mu and Viswanath, 2018), which achieves high scores on Word and Semantic textual similarity tasks. The algorithm removes the projections of

top principal components from each of the word vectors, making the individual word vectors more discriminative (Refer to Algorithm 1 for details).

---

### Algorithm 1: Post Processing Algorithm PPA(X, D)

---

**Data:** Embedding Matrix X, Threshold Parameter D  
**Result:** Post-Processed Word Embedding Matrix X  
1  $X = X - \bar{X}$ ; // Subtract Mean Embedding  
/\* Compute PCA Components \*/  
2  $u_i = \text{PCA}(X)$ , where  $i = 1, 2 \dots d$ ;  
/\* Remove Top-D Components \*/  
3 **for all**  $v$  **in**  $X$  **do**  
4  $v = v - \sum_{i=1}^D (u_i^T \cdot v) u_i$   
5 **end**

---

Table 3: BLEU scores over three different low-resource language pairs with pretrained embeddings and Top D components removed using PPA. **Green** cells denotes top scores.

	AZ->EN	BE->EN	GL->EN
<b>Pre-Trained</b>	<b>3.24</b>	<b>6.09</b>	<b>15.91</b>
<b>PPA (D = 1)</b>	3.19	6.02	14.81
<b>PPA (D = 2)</b>	3.07	5.50	13.88
<b>PPA (D = 3)</b>	3.04	5.26	13.27
<b>PPA (D = 4)</b>	2.92	4.75	13.24

### 5.1 Sentence Classification Tasks

We compare the performance of PPA (with a constant  $D=5$  across all the embeddings) on 9 downstream sentence classification tasks. Table 2 shows that such post-processing doesn’t always lead to gains in accuracy and can be counterproductive in a number of tasks. This suggests that within the context of downstream sentence classification tasks, projecting word vectors away from the top components leads to a loss of ‘useful’ information. This could again be explained using the analysis from Figure 3, wherein it is evident that the top dimensions also contain syntactic information, the loss of which adversely impacts downstream classification tasks, which by construction, benefit from both semantic and syntactic information. On the same tasks, we also observe a drop in sentence classification accuracy (2.37/1.99/3.94 average drop on word2vec/Glove/fasttext) using 150 dimensional embeddings obtained from PPA based dimensionality reduction (Raunak, 2017).

### 5.2 Machine Translation

Recently, (Qi et al., 2018) have shown that pre-trained embeddings lead to significant gains in performance for the translation of three low resource languages namely, Azerbaijani (AZ), Belarusian (BE) and Galician (GL) into English



(EN). Here, we demonstrate the impact of the post processing algorithm on machine translation (MT) tasks. We replicate the experimental settings of (Qi et al., 2018) and use a standard 1 layer encoder-decoder model with attention (Bahdanau et al., 2014) and a beam size of 5. Prior to training, we initialize the encoder with fast-text word embeddings (no other embeddings are publically available for these languages) trained on Wikipedia<sup>5</sup>. We then use PPA on the pre-trained embeddings and train again. Results in Table 3 show that removing the top principal component(s) leads to a consistent drop in BLEU scores across the three language pairs. This can be explained using the analysis from earlier section i.e. instead of strengthening the embeddings, removing the top components leads to a loss of ‘useful’ information for the Machine translation task.

## 6 Conclusion and Future Work

To conclude, besides elucidating redundancy in the word embedding space, we demonstrate that the variance explained by the word embeddings’ principal components is not a reliable proxy for the downstream utility of the corresponding representations and that the syntactic information captured by a principal component does not depend on the amount of variance it explains. Further, we show that variance based post-processing is not suitable for tasks which rely more on syntax, such as sentence classification and machine translation. Further, we wish to explore whether the geometric intuitions developed for word embeddings could be leveraged for contextualized embeddings such as ElMo (Peters et al., 2018) and BERT (Devlin et al., 2018).

## References

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

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*.

Manaal Faruqi, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

Manaal Faruqi and Chris Dyer. 2014. Community evaluation and exchange of word vectors at word-vectors.org. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24.

Ian T Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Nikola Mrkšić, Diarmuid OSéaghdha, Blaise Thomson, and pages=142–148 year=2016 Gašić, Milica and Rojas-Barahona, Lina and Su, Pei-Hao and Vandyke, David and Wen, Tsung-Hsien and Young, Steve, booktitle=Proceedings of NAACL-HLT. Counter-fitting word vectors to linguistic constraints.

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

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.

<sup>5</sup> <https://bit.ly/2WkHQ0Y>

- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Janani Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? *arXiv preprint arXiv:1804.06323*.
- Vikas Raunak. 2017. Effective dimensionality reduction for word embeddings. *NIPS 2017, workshop on Learning with Limited Labeled Data*.
- Zi Yin and Yuanyuan Shen. 2018. [On the dimensionality of word embedding](#). In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 895–906. Curran Associates, Inc.