Autoencoding Improves Pre-trained Word Embeddings

Masahiro Kaneko
Tokyo Metropolitan University, Japan
kaneko-masahiro@ed.tmu.ac.jp

Danushka Bollegala
University of Liverpool, UK
danushka@liverpool.ac.uk

Abstract

Prior work investigating the geometry of pre-trained word embeddings have shown that word embeddings to be distributed in a narrow cone and by centering and projecting using principal component vectors one can increase the accuracy of a given set of pre-trained word embeddings. We theoretically prove that this post-processing step involving centering and projecting using the largest principal component vectors is equivalent to applying a linear autoencoder to minimise the squared ℓ_2 reconstruction error. This result is in contrast to prior work (Mu and Viswanath, 2018) that propose to remove the top principal components from pre-trained embeddings. Moreover, we experimentally verify our theoretical claims and show that retaining the top principal components is indeed useful for improving pre-trained word embeddings, without requiring access to any additional linguistic resources or labelled data.

1 Introduction

Pre-trained word embeddings have been successfully used as features for representing input texts in many NLP tasks (Dhillon et al., 2015; Mnih and Hinton, 2009; Collobert et al., 2011; Huang et al., 2012; Mikolov et al., 2013; Pennington et al., 2014). Mu and Viswanath (2018) showed that the accuracy of pre-trained word embeddings can be further improved in a post-processing step, without requiring additional training data, by removing the mean of the word embeddings (*centering*) computed over the set of words (i.e. vocabulary) and projecting onto the directions defined by the principal component vectors, except for the top few principal components. Specifically, they showed that most pre-trained word embeddings are distributed in a narrow cone around the mean embedding vector and post-processing by centering and projection helps to reinstate isotropy in the embedding space. This post-processing operation has been proposed in various other contexts such as distributional (counting-based) word representations (Sahlgren et al., 2016) and sentence embeddings (Arora et al., 2017).

Independently, to the line of work described above, autoencoders have been widely used for fine-tuning pre-trained word embeddings such as for removing gender bias (Kaneko and Bollegala, 2019), meta-embedding (Bao and Bollegala, 2018), cross-lingual word embedding learning (Wei and Deng, 2017) and domain adaptation (Chen et al., 2012), to name a few. However, it is unclear whether better performance is obtained simply by applying an autoencoder (a self-supervised task, requiring no labelled data) on pre-trained word embeddings, without performing any task-specific fine-tuning (requires labelled data for the task).

A connection between principal component analysis (PCA) and linear autoencoders was first proved by Baldi and Hornik (1989), extending the analysis by Bourlard and Kamp (1988). We revisit this analysis and show that according to the theory one must *retain* the largest principal components instead of removing them as proposed by Mu and Viswanath (2018) in order to minimise the squared ℓ_2 reconstruction loss. Next, we experimentally show that by applying a non-linear autoencoder we can post-process a given set of pre-trained word embeddings and obtain more accurate word embeddings than by the method proposed by Mu and Viswanath (2018). Although Mu and Viswanath (2018) motivated the removal of largest principal components as a method to improve the isotropy of the word embeddings, our empirical findings show that by applying an autoencoders we can achieve a similar effect in terms of isotropy.

2 Autoencoding as Centering and PCA Projection

Let us consider a set of n-dimensional pre-trained word embeddings, $\{\boldsymbol{x}_1, \ldots, x_N\}$ for a vocabulary $\mathcal{V}$ consisting of N words. We post-process these pre-trained word embeddings using an autoencoder consisting of a $p(< n)$ dimensional single hidden layer, an encoder (defined by a matrix $\mathbf{W}_e \in \mathbb{R}^{n \times p}$ and a bias vector $\boldsymbol{b}_e \in \mathbb{R}^p$) and a decoder (defined by a matrix $\mathbf{W}_d \in \mathbb{R}^{p \times n}$ and a bias vector $\boldsymbol{b}_d \in \mathbb{R}^n$). Let $\mathbf{X} \in \mathbb{R}^{n \times N}$ be the embedding matrix, where word embeddings are arranged in columns. Using matrices $\mathbf{B} \in \mathbb{R}^{p \times N}$, $\mathbf{H} \in \mathbb{R}^{p \times N}$ and $\mathbf{Y} \in \mathbb{R}^{n \times N}$ respectively denoting the activations, hidden states and reconstructed output embeddings, the autoencoder can be specified as follows.

$$\mathbf{B} = \mathbf{W}_e\mathbf{X} + \boldsymbol{b}_e\boldsymbol{u}^\top \tag{1}$$

$$\mathbf{H} = F(\mathbf{B}) \tag{2}$$

$$\mathbf{Y} = \mathbf{W}_d\mathbf{H} + \boldsymbol{b}_d\boldsymbol{u}^\top \tag{3}$$

Here, $\boldsymbol{u} \in \mathbb{R}^N$ is a vector consisting of ones and F is an element-wise activation function. The squared ℓ_2 reconstruction loss, J, for the autoencoder is given by (4).

$$J(\mathbf{W}_e, \mathbf{W}_d, \boldsymbol{b}_e, \boldsymbol{d}_d) = \left|\left|\mathbf{W}_d F(\mathbf{W}_e\mathbf{X} + \boldsymbol{b}_d\boldsymbol{u}^\top) + \boldsymbol{b}_d\boldsymbol{u}^\top\right|\right|^2 \tag{4}$$

Lemma 1. *Let $\mathbf{X}'$ and $\mathbf{H}'$ respectively denote the centred embedding and hidden state matrices. Then,* (4) *can be expressed using $\mathbf{X}'$ and $\mathbf{H}'$ as $J(\mathbf{W}_e, \mathbf{W}_d, \boldsymbol{b}_d, \hat{\boldsymbol{b}}_d) = ||\mathbf{X}' - \mathbf{W}_d\mathbf{H}'||^2$, where the decoder's optimal bias vector is given by $\hat{\boldsymbol{b}}_d = \frac{1}{N}(\mathbf{X} - \mathbf{W}_d\mathbf{H})\,\boldsymbol{u}$.*

See the Supplementary for the proof.

Lemma 1 holds even for non-linear autoencoders and claims that the centering happens automatically during the minimisation of the reconstruction error. Following Lemma 1, we can assume that the embedding matrix, $\mathbf{X}$, to be already centred and can limit further discussions to this case. Moreover, after centering the input embeddings, the biases can be *absorbed* into the encoder/decoder matrices by setting an extra dimension that is always equal to 1 in the pre-trained word embeddings. This has the added benefit of simplifying the notations and proofs. Under these conditions Theorem 2 shows an important connection between linear autoencoders and PCA.

Theorem 2. *Assume that $\mathbf{\Sigma}_{xx} = \mathbf{X}\mathbf{X}^\top$ is full-rank with n distinct eigenvalues $\lambda_1 > \ldots > \lambda_n$. Let $\mathcal{I} = \{i_1, \ldots, i_p\}$ $(1 \leq i_1 < \ldots < i_p \leq n)$ be any ordered p-index set, and $\mathbf{U}_\mathcal{I} = [\boldsymbol{u}_{i_1}, \ldots \boldsymbol{u}_{i_p}]$ denote the matrix formed by the orthogonal eigenvectors of $\mathbf{\Sigma}_{xx}$ associated with the eigenvalues $\lambda_{i_1}, \ldots, \lambda_{i_p}$. Then, two full-rank matrices $\mathbf{W}_d$ and $\mathbf{W}_e$ define a critical point of* (4) *for a linear autoencoder if and only if there exists an ordered p-index set $\mathcal{I}$ and an invertible matrix $\mathbf{C} \in \mathbb{R}^{p \times p}$ such that*

$$\mathbf{W}_d = \mathbf{U}_\mathcal{I}\mathbf{C} \tag{5}$$

$$\mathbf{W}_e = \mathbf{C}^{-1}\mathbf{U}_\mathcal{I}^{-1}. \tag{6}$$

Moreover, the reconstruction error, $J(\mathbf{W}_e, \mathbf{W}_d)$ can be expressed as

$$J(\mathbf{W}_e, \mathbf{W}_d) = \mathrm{tr}(\mathbf{\Sigma}_{xx}) - \sum_{t \in \mathcal{I}} \lambda_t. \tag{7}$$

Proof of Theorem 2 and approximations for non-linear activations are given in the supplementary.

Because $\mathbf{\Sigma}_{xx}$ is a covariance matrix, it is positive semi-definite. Strict positivity corresponds to it being full-rank and is usually satisfied in practice for pre-trained word embeddings, which are dense and use a small $n(\ll N)$ independent dimensions for representing the semantics of the words. Moreover, $\mathbf{W}_e$, $\mathbf{W}_d$ are randomly initialised in practice making them full-rank as assumed in Theorem 2.

The connection between linear autoencoders and PCA was first proved by Baldi and Hornik (1989), extending the analysis by Bourlard and Kamp (1988). Reconstructing the principal component vectors from an autoencoder has been discussed by Plaut (2018) without any formal proofs. However, to the

best of our knowledge, a theoretical justification for post-processing pre-trained word embeddings by autoencoding has not been provided before.

According to Theorem 2, we can minimise (7) by selecting the largest eigenvalues as λ_t. This result contradicts the proposal by Mu and Viswanath (2018) to project the word embeddings away from the largest principal component vectors, which is motivated as a method to improve isotropy in the word embedding space. They provided experimental evidence to the effect that largest principal component vectors encode word frequency and removal of it is not detrimental to semantic tasks such as semantic similarity measurement and analogy detection. However, the frequency of a word is an important piece of information for tasks that require differentiating stop words and content words such as in information retrieval. Moreover, contextualised word embeddings such as BERT (Devlin et al., 2019) and Elmo (Peters et al., 2018) have shown to be anisotropic despite their superior performance in a wide-range of NLP tasks (Ethayarajh, 2019). Therefore, it is not readily obvious whether removing the largest principal components to satisfy isotropy is a universally valid strategy. On the other hand, our experimental results show that by autoencoding not only we obtain better embeddings than Mu and Viswanath (2018), but also it improves the isotropy of the pre-trained word embeddings.

3 Experiments

To evaluate the proposed post-processing method, we use the following pre-trained word embeddings: **Word2Vec**[1] (300-dimensional embeddings for 3,000,000 words learned from the Google News corpus), **GloVe**[2] (300-dimensional word embeddings for 2,196,016 words learned from the Common Crawl), and **fastText**[3] (300-dimensional embeddings for 999,994 words learned from Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset). We input each set of embeddings separately to an autoencoder with one hidden layer and minimise the squared ℓ_2 error using ADAM as the optimiser. After training, the pre-trained embeddings are sent through the trained autoencoder and the hidden layer outputs are used as the post-processed word embeddings. Due to space limitations, we report results for an autoencoder (denoted as **AE**) with 300-dimensional hidden layer and a $\tanh$ activation in the paper (other results are shown in the supplementary). We compare the embeddings post-processed by the method proposed by Mu and Viswanath (2018), which removes the top principal components from the pre-trained embeddings. We denote this method by **ABTT**[4].

Table 1 compares the performance of the **original** embeddings vs. embeddings obtained after post-processing by **ABTT** and the proposed method, **AE**. For the semantic similarity benchmarks, a high degree of Spearman correlation between human similarity ratings and the cosine similarity scores computed using the word embeddings is considered as better. From Table 1 we see that **AE** improves word embeddings and outperforms **ABTT** in almost all semantic similarity datasets. For the word analogy benchmarks, we use the PairDiff method (Levy and Goldberg, 2014) to predict the fourth word to complete a proportional analogy and the accuracy of the prediction is reported. On word analogy tasks, we see that for GloVe embeddings **AE** reports the best performance but **ABTT** performs better for fastText. Overall, the improvement due to post-processing is less prominent in the word analogy task. This was also reported by Mu and Viswanath (2018) and is explained by the fact that analogy solving is done using vector difference, which is not impacted by centering. In the concept categorisation task, we measure the Euclidean distance between two words computed using their embeddings as the distance measure and conduct use k-means clustering a to group words into clusters separately in each benchmark dataset. Cluster purity (Manning et al., 2008) is computed as the evaluation measure using the gold category labelling provided in the benchmark datasets. High values of purity would indicate that the word embeddings capture information related to the semantic classes of words. From Table 1 we see that AE outperforms ABTT in all cases, except on BLESS with Word2Vec embeddings.

Following the definition given by Mu and Viswanath (2018), we empirically estimate the isotropy of a

[1] `https://code.google.com/archive/p/word2vec/`

[2] `https://github.com/stanfordnlp/GloVe`

[3] `https://fasttext.cc/docs/en/english-vectors.html`

[4] Stands for *all-but-the-top*

Embedding	Word2Vec			GloVe			fastText		
Dataset	original	ABTT	AE	original	ABTT	AE	original	ABTT	AE
WS-353	62.4	61.2	**61.8**	60.6	61.5	**65.8**	65.9	67.7	**69.0**
SIMLEX-999	44.7	45.4	**45.5**	39.5	41.5	**42.2**	46.2	47.4	**48.8**
RG-65	75.4	76.0	**76.3**	68.1	68.0	**72.3**	78.4	**81.4**	80.5
MTurk-287	**69.0**	68.9	68.9	71.8	71.9	**74.4**	73.3	73.8	**74.7**
MTurk-771	63.1	63.7	**63.9**	62.7	63.7	**67.7**	69.6	71.8	**72.4**
MEN	68.1	68.3	**69.3**	67.7	69.5	**74.8**	71.1	75.7	**76.0**
MSR	**73.6**	73.2	73.4	73.8	73.2	**74.4**	87.1	**88.0**	87.3
Google	74.0	**74.8**	74.3	76.8	76.9	**77.1**	85.3	**88.0**	86.4
SemEval	20.0	19.9	**20.3**	15.4	17.2	**17.6**	21.0	23.2	**23.3**
BLESS	70.5	**71.0**	70.0	76.5	76.5	**79.5**	75.5	79.0	**80.5**
ESSLI	75.5	73.7	**76.2**	72.2	72.2	**73.0**	74.7	76.2	**77.0**

Table 1: Performance on semantic similarity (**WS-353** (Agirre et al., 2009), **SIMLEX-999** (Hill et al., 2015), **RG-65** (Rubenstein and Goodenough, 1965), **MTurk-287** (Radinsky et al., 2011), **MTurk-771** (Halawi et al., 2012) and **MEN** (Bruni et al., 2014)), analogy (**Google**, **MSR** (Mikolov et al., 2013), and **SemEval** (Jurgens et al., 2012)) and concept categorisation (**BLESS** (Baroni and Lenci, 2011) and **ESSLI** (Baroni et al., 2008)) benchmarks. Results are shown for the original embeddings and their post-processed versions by ABTT and the proposed autoencoder (**AE**) for pre-trained Word2Vec, GloVe and fastText embeddings.

	Original	ABTT	AE
Word2Vec	0.489	0.981	0.976
GloVe	0.018	0.943	0.884
fastText	0.773	0.995	0.990

Table 2: The measure of isotropy of original and post-processed using ABTT and AE embeddings.

set of embeddings as $\frac{\min_{\boldsymbol{c}\in\mathcal{C}} Z(\boldsymbol{c})}{\max_{\boldsymbol{c}\in\mathcal{C}} Z(\boldsymbol{c})}$, where $\mathcal{C}$ is the set of principal component vectors computed for the given set of pre-trained word embeddings and $Z(\boldsymbol{c}) = \sum_{x\in\mathcal{V}} \exp(\boldsymbol{c}^\top \boldsymbol{x})$ is the normalisation coefficient in the partition function defined in (Arora et al., 2016). From Table 2 we see that compared to the original embeddings both **ABTT** and **AE** improves isotropy. However, unlike **ABTT**, which explicitly removes the top principal components to improve isotropy, we see that **AE** automatically improves isotropy by autoencoding.

In addition to the theoretical and empirical advantages of autoencoding as a post-processing method, it is practically attractive operation as well. Unlike PCA, which must be computed using embeddings for all the words in the vocabulary as a batch operation, autoencoders could be run in an online fashion using only a small mini-batch of words at a time. Moreover, non-linear transformations and regularisation (e.g. in the from of dropout) can be easily incorporated into autoencoders, which can also be stacked for further post-processing. While online (Warmuth and Kuzmin, 2007; Feng et al., 2013a; Feng et al., 2013b) and non-linear (Scholz et al., 2005) variants of PCA have been proposed, they have not been popular among practitioners due to their complexity, scalability and lack of availability in deep learning frameworks.

4 Conclusion

We showed that applying a non-linear autoencoder improves pre-trained word embeddings and outperforms the prior proposal for removing top principal components. We provided theoretical justifications to our proposal and empirically showed that it also improves isotropy of the word embeddings.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human*

Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 19–27. Association for Computational Linguistics.

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.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proc. of ICLR*.

Pierre Baldi and Kurt Hornik. 1989. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, Jan.

Cong Bao and Danushka Bollegala. 2018. Learning word meta-embeddings by autoencoding. In *Proc. of the 27th International Conference on Computational Linguistics (COLING)*, pages 1650–1661.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *GEMS'11 Workshop on Models of Natural Language Semantics*.

Marco Baroni, Stefan Evert, and Alessandro Lenci. 2008. Esslli workshop on distributional lexical semantics bridging the gap between semantic theory and computational simulations.

H. Bourlard and Y. Kamp. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, Sep.

E. Bruni, N. K. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, Jan.

Minmim Chen, Zhixiang (Eddie) Xu, and Kilian Q. Weinberger. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proc. of ICML*.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuska. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493 – 2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Paramveer S. Dhillon, Dean P. Foster, and Lyle H. Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research*, 16:3035–3078.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November. Association for Computational Linguistics.

Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. 2013a. Online pca for contaminated data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 764–772. Curran Associates, Inc.

Jiashi Feng, Huan Xu, and Shuicheng Yan. 2013b. Online robust pca via stochastic optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 404–412. Curran Associates, Inc.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proc. of KDD*, pages 1406–1414.

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

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL'12*, pages 873 – 882.

David A. Jurgens, Saif Mohammad, Peter D. Turney, and Keith J. Holyoak. 2012. Measuring degrees of relational similarity. In *Proc. of SemEval*.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650, Florence, Italy, July. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *CoNLL*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representation in vector space. In *Proc. of International Conference on Learning Representations*.

Andriy Mnih and Geoffrey E. Hinton. 2009. A scalable hierarchical distributed language model. In *Proc. of NIPS*, pages 1081–1088.

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

Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: global vectors for word representation. In *Proc. of EMNLP*, pages 1532–1543.

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

Elad Plaut. 2018. From Principal Subspaces to Principal Components with Linear Autoencoders.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: Computing word relatedness using temporal semantic analysis. In *WWW'11*, pages 337 – 346.

H. Rubenstein and J.B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8:627–633.

Magnus Sahlgren, Amaru Cuba Gyllensten, Fredrik Espinoza, Ola Hamfors, Jussi Karlgren, Fredrik Olsson, Per Persson, Akshay Viswanathan, and Anders Holst. 2016. The gavagai living lexicon. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 344–350, Portorož, Slovenia, May. European Language Resources Association (ELRA).

M. Scholz, F. Kaplan, C. L. Guy, J. Kopka, and J. Selbig. 2005. Non-linear pca: a missing data approach. *Bioinformatics*, 21(20):3887–3895, Aug.

Manfred K. K Warmuth and Dima Kuzmin. 2007. Randomized pca algorithms with regret bounds that are logarithmic in the dimension. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1481–1488. MIT Press.

Liangchen Wei and Zhi-Hong Deng. 2017. A variational autoencoding approach for inducing cross-lingual word embeddings. In *Proc. of IJCAI*, pages 4165–4171.