RPD: A Distance Function Between Word Embeddings

Xuhui Zhou,[1] **Zaixiang Zheng,**[2] **Shujian Huang,**[2]
[1]University of Washington
[2]National Key Laboratory for Novel Software Technology, Nanjing University
xuhuizh@uw.edu, zhengzx@smail.nju.edu.cn, huangsj@nju.edu.cn

Abstract

It is well-understood that different algorithms, training processes, and corpora produce different word embeddings. However, less is known about the relation between different embedding spaces, i.e. how far different sets of embeddings deviate from each other. In this paper, we propose a novel metric called Relative pairwise inner Product Distance (RPD) to quantify the distance between different sets of word embeddings. This metric has a unified scale for comparing different sets of word embeddings. Based on the properties of RPD, we study the relations of word embeddings of different algorithms systematically, and investigate the influence of different training processes and corpora. The results shed light on the poorly understood word embeddings and justify RPD as a measure of the distance of embedding spaces.

1 Introduction

Word embeddings are important in Natural language processing (NLP) which map words into a low-dimensional vector space. Many works have been proposed to generate word embeddings (Mnih and Kavukcuoglu, 2013; Mikolov et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014a; Bojanowski et al., 2017; Devlin et al., 2019).

With many different sets of word embeddings produced by different algorithms and corpora, it is interesting to investigate the relationships between these sets of word embeddings. Intrinsically, this would help us better understand word embeddings (Levy et al., 2015). Practically, knowing the relationship between different sets of word embeddings helps us build better word meta-embeddings (Yin and Schütze, 2016), reduce biases in word embeddings (Bolukbasi et al., 2016), pick better hyperparameters (Yin and Shen, 2018), and choose suitable algorithms in different scenarios (Kozlowski et al., 2019).

To study the relationship between different embedding spaces systematically, we propose RPD as a measure of the distance between different sets of embeddings. We derive statistical properties of RPD including its asymptotic upper bound and normality under the independence condition. We also provide a geometric interpretation of RPD. Furthermore, we show that RPD is strongly correlated with the performance of word embeddings measured by intrinsic metrics, such as comparing semantic similarity and evaluating analogies.

With the help of RPD, we study the relations among several popular embedding methods, including GloVe (Pennington et al., 2014), SGNS[1] (Mikolov et al., 2013), Singular Value Decomposition (SVD) factorization of PMI matrix, and SVD factorization of log count (LC) matrix. Results show that these methods are statistically correlated, which suggests that there is an unified theory behind these methods.

Additionally, we analyze the influence of training processes, i.e. hyperparameters (negative sampling), random initialization; and the influence of corpora towards word embeddings. Our findings include the fact that different training corpora result in significantly different GloVe embeddings, and that the main difference between embedding spaces comes from the algorithms although hyperparameters also have certain influence. Those findings not only provide some interesting insights of word embeddings but also fit nicely with our intuition, which further proves RPD as a suitable measure to quantify the relationship between different sets of word embeddings.

2 Background

Before introducing RPD, we review the theory behind some static word embedding methods, and

[1] Skip-gram with Negative Sampling

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 42–50
July 5 - July 10, 2020. ©2020 Association for Computational Linguistics

discuss some previous works investigating the relationship between embedding spaces.

2.1 Word Embedding Models

We consider the following four word embedding models: SGNS, GloVe, SVD_{PMI}, SVD_{LC}. SGNS and GloVe are two widely used embedding methods, while SVD_{PMI} and SVD_{LC} are matrix factorization methods which are intrinsically related to SGNS and GloVe (Levy and Goldberg, 2014b; Levy et al., 2015; Yin and Shen, 2018).

The embedding of all the words forms an embedding matrix $E \in \mathbf{R}^{n \times d}$, where the d here is the dimension of each word vector and n is the size of the vocabulary.

SGNS maximizes a likelihood function for word and context pairs that occur in the dataset and minimizes it for randomly sampled unobserved pairs, i.e. negative samples (NS). We denote the method with k NS as SGNS_k.

GloVe factorizes the log-count matrix shifted by the entire vocabulary's bias term. The bias here are parameters learned stochastically with an objective weighted according to the frequency of words.

$\textbf{SVD}_{\textbf{PMI/LC}}$ SVD factorizes a signal matrix $M = UDV^T$, which aims at reducing the dimensions of the cooccurrence matrix. The resulting embedding is $E = U_{:,1:d} D^{\frac{1}{2}}_{1:d,1:d}$, where d is the dimension of word embeddings. We denote the method as SVD_{PMI}, if the signal is the PMI matrix, and SVD_{LC} if the signal is the log count matrix.

Although the scope of this paper focuses on standard word embeddings that were learned at the word level, RPD could be adapted to analyze embeddings that were learned from word pieces, for example, fastText (Bojanowski et al., 2017) and contextualized embeddings (Peters et al., 2018; Devlin et al., 2019).

2.2 Relationship Between Embedding Spaces

Levy and Goldberg (2014b) provide a good analogy between SGNS and SVD_{PMI}. They suggest that SGNS is essentially factorizing the pointwise mutual information (PMI) matrix. However, their analogy is based on the assumption of no dimension constraint in SGNS, which is not possible in practice. Furthermore, their analogy is not suitable for analyzing methods besides SGNS and PMI models since their theoretical derivation relies on the specific objective of SGNS.

Yin and Shen (2018) provide a way to select the best dimension of word embeddings for specific tasks by exploring the relations of embedding spaces of different dimension. They introduce Pairwise Inner Product (PIP) loss (Yin and Shen, 2018), an unitary-invariant metric for measuring word embeddings' distance (Smith et al., 2017). The unitary-invariance of word embeddings states that two embedding vector spaces are equivalent if one can be obtained from another by multiplying a unitary matrix. However, PIP loss is not suitable for comparing numerically across embedding spaces since PIP loss has different energy for different embedding spaces.

3 Quantifying Distances between Embeddings

In this section, we describe the definition of RPD and its properties, which make RPD a suitable and effective method to quantify the distance between embedding spaces. Note that two embedding spaces do not necessarily have the same vocabulary for calculating the RPD.

3.1 RPD

For the following discussion, we always use the Frobenius norm as the norm of matrices.

Definition 1. (RPD) The RPD between embedding matrices E_1 and E_2 is defined as follows:

$$\text{RPD}(E_1, E_2) = \frac{1}{2} \frac{\|\tilde{E}_1 \tilde{E}_1^{\,T} - \tilde{E}_2 \tilde{E}_2^{\,T}\|^2}{\|\tilde{E}_1 \tilde{E}_1^{\,T}\| \|\tilde{E}_2 \tilde{E}_2^{\,T}\|}.$$

where $\tilde{E}$ comes from dividing each entry of E by its standard deviation. For convenience, we let $\tilde{E} \equiv E$ for the following discussion.

The numerator of RPD respects the unitary-invariant property of word embeddings, which means that unitary transformation (i.e. rotation) preserves the relative geometry of an embedding space. The denominator is a normalization, which allows us to regard the whole embedding matrix as an integrated part (i.e. RPD does not correlate with the number of words of embedding spaces). This step makes comparisons across methods possible.

3.2 Statistical Properties of RPD

We assume the widely used isotropic assumption (Arora et al., 2016) that the ensemble of word vectors consists of i.i.d draws generated by $v = s\hat{v}$, where $\hat{v}$ is from the spherical Gaussian distribution, and s is a scalar random variable. In our case, we can assume each entry of embedding comes from a standard normal distribution E: $v_{ij} \sim \mathcal{N}(0, 1)$.

Note that the assumption may not always work in practice, especially for other embeddings such as contextualized embeddings. However, under the isotropic conditions, the statistical properties derived are intuitively and empirically plausible. Besides, those properties serve to better interpret the value of RPD alone. Since RPD, in many cases, is used for comparison, we should be comfortable with the assumption.

Upper bound We estimate the asymptotic upper bound of RPD. By factorizing the numerator of RPD, we get (1).

$$\begin{aligned}\mathrm{RPD}(E_1, E_2) = \frac{1}{2}\frac{\|E_1E_1^T\|^2 + \|E_2E_2^T\|^2}{\|E_1E_1^T\|\|E_2E_2^T\|} \\ - \frac{\langle E_1E_1^T, E_2E_2^T\rangle}{\|E_1E_1^T\|\|E_2E_2^T\|} \quad (1)\end{aligned}$$

Applying the Cauchy-Schwarz inequality to the last term of (1)[2], we have the following estimation.

$$\begin{aligned}2\mathrm{RPD}(E_1, E_2) &\leq \frac{\|E_1E_1^T\|^2 + \|E_2E_2^T\|^2}{\|E_1E_1^T\|\|E_2E_2^T\|} \\ &= \frac{\|E_1E_1^T\|}{\|E_2E_2^T\|} + \frac{\|E_2E_2^T\|}{\|E_1E_1^T\|}\end{aligned} \quad (2)$$

By the law of large numbers, we can prove that $\lim_{n\to\infty} \|EE^T\| = n\sqrt{d}$ (Appendix A). Then, we can tell from (2) that RPD is bounded by 1 when $n \to \infty$. In practice, the number of words n is large enough to let the maximum of RPD stay around 1, which means RPD is well-defined numerically.

Normality For $\mathrm{RPD}(E_1, E_2)$, if E_1 is independent of E_2, we can prove that RPD distributes normally from both an empirical and a theoretical perspective. Theoretically, by applying the central limit theorem to the numerator and the law of large numbers to the denominator of RPD, we can get the normality of RPD under the condition $n \to \infty$, $\frac{d}{n} = c$, where c remains constant (Appendix B). Empirically, we can use Monte Carlo simulation to show the normality and estimate the mean and variance of RPD (Appendix C). With the help of RPD, we can perform hypothesis test (z-test) to evaluate the independence of two embedding spaces.

3.3 Geometric Interpretation of RPD

From equation (1), we can tell that the first term goes to 1 when $n \to \infty$. So we only need to discuss the second term.

$$\frac{\langle E_1E_1^T, E_2E_2^T\rangle}{\|E_1E_1^T\|\|E_2E_2^T\|}$$

For the i^{th} row in EE^T, we have vector $\hat{v}_i = (v_iv_1^T, v_iv_2^T, ..., v_iv_n^T)$, where v_i is the word i's vector in embedding E, n is the number of words. We can interpret $\hat{v}_i$ as another representation of word i projected onto the space spanned by $v_1, v_2, ..., v_n$. So for convenience, we denote $\hat{E} = EE^T$ with its i^{th} row as $\hat{v}_i$.

We can prove that $\lim_{n\to\infty} \mathrm{RPD}(E_1, E_2) = 1 - \frac{1}{n}\sum_{i=1}^{n}\cos(\theta_i)$. The $\theta_i \in (0, \frac{\pi}{2})$ is the angle between $\hat{v}_i^{(1)}$ (i^{th} row vector of $\hat{E}_1$) and $\hat{v}_i^{(2)}$ (i^{th} row vector of $\hat{E}_2$) (Appendix D). Therefore, we can understand the value of RPD from the perspective of cosine similarity between vectors.

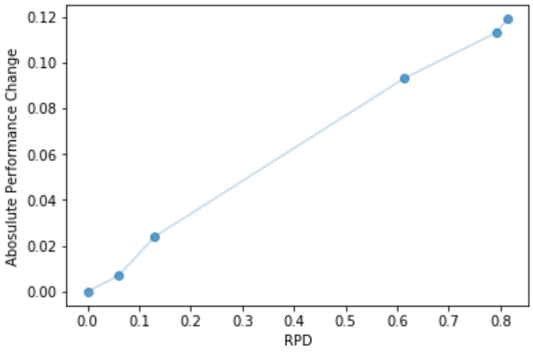

Figure 1: The plot shows the difference in performance as a function of RPD score. The x-axis for each point represents the RPD between word embeddings produced by SGNS (with NS 15, 5, 1), GloVe, SVD$_{\mathrm{PMI}}$, SVD$_{\mathrm{LC}}$ and word embeddings produced by SGNS$_{25}$. The y-axis for each point represents the sum of absolute variation in the performance (word similarity and word analogy).

3.4 RPD and Performance

As Yin and Shen (2018) discussed, usability of word embeddings, such as using them to solve analogy and relatedness tasks, is important to practitioners. Through applying different sets of word embeddings to word similarity and word analogy tasks (Mikolov et al., 2013), we study the relationship between RPD and word embeddings' performance. Specifically, we set the word embeddings produced by SGNS with 25 NS as a starting point and use

[2] The inner product of matrix A and B is defined as $\langle A, B\rangle = trace(A^TB)$

other word embeddings, for example, GloVe as an end point. Then we get a two dimensional point with x as their RPD, y as their absolute performance change in word similarity[3] and analogy[4] tasks.

By putting those points in Figure 1, we can tell in a certain range of RPD, the larger RPD between the two sets of word embedding means the bigger gap in their absolute performance. Intuitively, RPD is strongly related to cosine similarity, which is the measure of word similarity. RPD also shares the same property of PIP loss, where a small RPD leads to a small difference in relatedness and analogy tasks. We obtain similar results when the starting point is a different embedding space.

Note that this section serves to demonstrate the performance (at least in word similarity and analogy tasks) variation of different embedding spaces is correlated with their RPD. While we are aware of the relevance of other downstream tasks, we do not explore further since our focus lies in investigating the intrinsic geometry relation of embedding spaces.

4 Experiment

The following experiments serve to apply RPD to explore some questions of interest and further demonstrate that RPD is suitable for investigating the relations between embedding spaces. We leave applying RPD to help improve specific NLP tasks to future research. For example, RPD could be used for combining different embeddings together, which could help us produce better meta-embeddings (Kiela et al., 2018).

4.1 Setup

If not explicitly stated, the experiments are performed on Text8 corpus (Mahoney, 2011), a standard benchmark corpus used for various natural language tasks (Yin and Shen, 2018). For all methods we experiment, we train 300 dimension embeddings, with window size of 10, and normalize the embedding matrices with their standard deviation[5]. The default NS for SGNS is 15.

Methods	GloVe	SVD_{PMI}	SVD_{LC}
SGNS_{25}	0.792	0.609	0.847
SGNS_{15}	0.773	0.594	0.837
SGNS_{5}	0.725	0.550	0.805
SGNS_{1}	0.719	0.511	0.799

Table 1: RPDs of SGNS vs other methods

4.2 Different Algorithms Produce Different Embeddings

Dependence of SGNS and SVD_{PMI}

As discussed in the introduction, the relationship between embeddings trained with SGNS and SVD_{PMI} remains controversial (Arora et al., 2016; Mimno and Thompson, 2017). We use the results we obtain in Section 3.2 to test their dependence. For example, if one believes that E_1 trained with SGNS and E_2 trained with SVD_{PMI} have no relationship, then the null hypothesis H_0 would be: E_1 and E_2 are independent.

Under H_0, $\text{RPD}(E_1, E_2)$ asymptotically follows $\mathcal{N}(\mu, \sigma^2)$. Then the test statistic z is calculated as follows.

$$z = \frac{\text{RPD}(E_1, E_2) - \mu}{\sigma}$$

In our case, we estimate $\mu = 0.953$ and $\sigma = 0.001$ with Monte Carlo simulation with randomly initialized embeddings. Take $\text{RPD}(\text{E}_{\text{SGNS}_1},\text{E}_{\text{SVD}_{\text{PMI}}}) = 0.511$ from Table 1 as an example, the statistic $z = 442$, which means the p-value $\ll 0.01$. Thus, we can confidently reject H_0. Notice that we can test any two sets of word embeddings with this method. It is not hard to see that no pair of word embeddings in Table 1 are independent, which suggests that there exists an unified theory behind these methods.

SGNS is Closest to SVD_{PMI}

With the help of RPD, it is also interesting to investigate distances between embeddings produced by different methods. Here, we calculate the RPDs among SGNS (with negative sampling 25, 15, 5, 1), GloVe, SVD_{PMI}, SVD_{LC}.

Table 1 shows the RPDs between SGNS with different negative sampling numbers and other methods. From the table, we can tell that SGNS stays close to SVD_{PMI}, which confirms Levy and Goldberg (2014b)'s theory.

[3] Our word similarity task can be found here: https://aclweb.org/aclwiki/WordSimilarity-353_Test_Collection_(State_of_the_art)

[4] Our word analogy task can be found here: https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art)

[5] The code can be found on Bitbucket: https://bitbucket.org/omerlevy/hyperwords

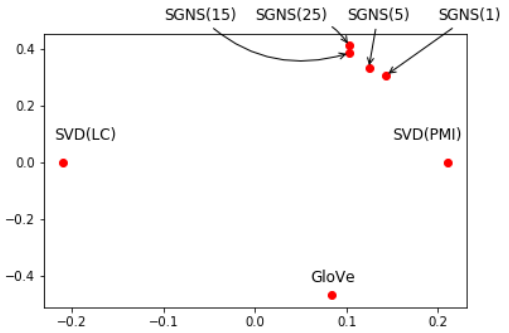

Figure 2: Plot of different methods. We create the plot by fixing the position of SVD_{LC} and SVD_{PMI}. We then derive the position of other word embeddings according to their RPD with existing points on the plot.

Hyper-parameters Have Influence on Embeddings

From Table 1, an interesting phenomenon is that SGNS becomes closer to other methods with the decrease of negative samples, which suggests that negative sampling is one of the factors driving SGNS away from matrix factorization methods.

With RPDs between different sets of word embeddings, we plot the embeddings in 2D by treating each embedding space as a single point. We first fix point SVD_{PMI} and SVD_{LC}, then we draw other points according to their RPDs with the other methods. Figure 2 helps us see how negative sampling affects the embedding intuitively. Increasing the number of negative samples pulls SGNS away from SVD_{PMI}. Combining Table 1 and Figure 2, we can tell that although the hyper-parameters can influence the embeddings to some extent, the main difference comes from the algorithms.

4.3 Different Initializations Barely Influence Embeddings

Random initializations produce different embeddings with the same algorithms and hyperparameters. While those embeddings usually get similar performance on the downstream tasks, people are still concerned about their effects. We investigate the influence of random initializations for GloVe and SGNS.

We train the embedding in the same setting multiple times and get the average RPDs for each method. For SGNS, the average RPDs of random initialization is 0.027. For GloVe, the value is 0.059.

We can tell that different random initializations produce essentially the same embeddings. Neither SGNS or GloVe has a significant RPD in different initializations, which suggests random initialization has little influence over word embeddings' performance (Section 3.4). However, SGNS seems to be more stable in this setting.

	SGNS	GloVe
Text8-WMT14	0.168	0.686
Text8-TED	0.119	0.758
WMT14-TED	0.175	0.716

Table 2: RPDs between same method trained from different corpora

4.4 Different Corpora Produce Different Embeddings

It is well known that different corpora produce different word embeddings. However, it is hard for us to tell how different they are and whether the difference influences downstream applications (Antoniak and Mimno, 2018). Knowing this would help researchers choose the algorithms in specific scenarios, for example, evolving semantic discovery (Yao et al., 2018; Kozlowski et al., 2019). They focus on the semantic evolution of words, but corpora are different in different time scales. Their methods use word embeddings to study semantic shift, which might be influenced by the word embeddings being trained on different corpora, thus getting unreliable results. In this case, it would be important to chose an algorithm less prone to influences by differences in corpora.

We train word embeddings using each of text8 (Wikipedia domain, 25097 unique words), WMT14 news crawl[6] (Newswire domain, 24359 unique words), TED speech[7] (Speech domain, 7389 unique words). We compute RPD on the intersections of their vocabulary

From Table 2, we can tell that SGNS is consistently more stable than GloVe in different domains. We suggest that this is because GloVe trains the embedding with co-occurrence matrix, which gets influenced more by the corpus.

5 Discussion

While our work investigates some interesting problems about word embeddings, there are many other

[6] http://www.statmt.org/wmt14/
[7] https://workshop2016.iwslt.org/

problems about embeddings that can be demonstrated with the help of RPD. We discuss some of them as follows.

5.1 RPD and Crosslingual Word Embeddings

Artetxe et al. (2018) provide a framework to obtain bilingual embeddings, whose the core step of the framework is an orthogonal transformation and other existing methods can be seen as its variations. The framework proposes to train monolingual embeddings separately and then map them into a shared-embedding space with linear transformation.

While linear transformation is no guarantee for the alignment of two embedding spaces from different languages, RPD could potentially serve as a way to indicate how different language pairs benefit from mapping them with an orthogonal transformation. Since RPD is unitary-invariant, we can calculate RPD between embedding spaces from different language pairs. The smaller RPD is, the better the framework could align this two language embedding spaces.

5.2 RPD and Post-Processing Word Embeddings

Post-processing word embeddings can be useful in many ways. For example, Vulić et al. (2018) retrofit word embeddings with external linguistic resources, such as WordNet to obtain better embeddings; Rothe and Schütze (2016) decompose embedding space to get better performance at specialized domains; and Mu and Viswanath (2018) obtain stronger embeddings by eliminating the common mean vector and a few top dominating directions.

RPD could serve as a metric to evaluate how the embedding space changes intrinsically after post-processing.

5.3 RPD and Contextualized Word Embeddings

Contextualized embeddings are popular NLP techniques which significantly improve a wide range of NLP tasks (Bowman et al., 2015; Rajpurkar et al., 2018). To understand why contextualized embeddings are beneficial to those NLP tasks, many works investigate the the nature of syntactic (Liu et al., 2019), semantic (Liu et al., 2019), and commonsense knowledge (Zhou et al., 2019) contained in such representations.

However, we still know little about the vector space of contextualized embeddings and their relationship with traditional word embeddings, which is important to further apply contextualized embeddings in various scenarios (Lin and Smith, 2019). RPD can potentially serve to help us better understand contextualized embeddings in future research.

6 Conclusion

In this paper, we propose RPD, a metric to quantify the distance between embedding spaces (i.e different sets of word embeddings). With the help of RPD and its properties, we verify some intuitions and answer some questions. Justifying RPD theoretically and empirically, we believe RPD can offer us a new perspective to understand and compare word embeddings.

Acknowledgments

I would like to thank Dr. Zi Yin, Dr. Vered Shwartz, Maarten Sap, and Jorge Balazs for their feedback that greatly improved the paper.

References

Maria Antoniak and David Mimno. 2018. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

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.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations.

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

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4349–4357. Curran Associates, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

632–642, Lisbon, Portugal. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477, Brussels, Belgium. Association for Computational Linguistics.

Austin C. Kozlowski, Matt Taddy, and James A. Evans. 2019. The geometry of culture: Analyzing the meanings of class through word embeddings. *American Sociological Review*, 84(5):905–949.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2177–2185, Cambridge, MA, USA. MIT Press.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Lucy H. Lin and Noah A. Smith. 2019. Situating sentence embedders with nearest neighbor overlap. *ArXiv*, abs/1909.10724.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Mahoney. 2011. Large text comparison benchmark, 2011.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc.

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, Doha, Qatar. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Sascha Rothe and Hinrich Schütze. 2016. Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–517, Berlin, Germany. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR*, abs/1702.03859.

Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Post-specialisation: Retrofitting vectors of words unseen in lexical resources. 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 516–527, New Orleans, Louisiana. Association for Computational Linguistics.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM 18, page 673681, New York, NY, USA. Association for Computing Machinery.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

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 887–898. Curran Associates, Inc.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2019. Evaluating commonsense in pretrained language models.

A Appendix A. Limitation of $||EE^T||$

As discuss before, in our case, we can assume i.i.d. $v_{ij} \sim \mathcal{N}(0,1)$, where v_{ij} is the j^{th} entry in the i^{th} word vector v_i of E.

$$\begin{aligned} ||EE^T|| &= n\sqrt{\frac{\sum_{i,j}^n (v_i v_j^T)^2}{n^2}} \\ &= n\sqrt{\frac{\sum_{i\neq j}^n (v_i v_j^T)^2}{n^2} + \frac{\sum_{i=j}^n (v_i v_j^T)^2}{n^2}} \end{aligned} \tag{3}$$

By the assumption, we know that $v_i v_j^T$ identically distributes for any $i \neq j, 1 \geq i \leq n, 1 \geq j \leq n$. By applying the law of large numbers, the term $\frac{\sum_{i\neq j}^n (v_i v_j^T)^2}{n^2}$ goes to $E((v_i v_j^T)^2)$ as n goes to ∞. The term $\frac{\sum_{i=j}^n (v_i v_j^T)^2}{n^2}$ goes to zero as n goes to ∞. Then, we know that $||EE^T|| \rightarrow n\sqrt{E((v_i v_j^T)^2)}, n \rightarrow \infty$.

We only need to calculate $E((v_i v_j^T)^2)$.

$$E((v_i v_j^T)^2) = Var(v_i v_j^T) + (E(v_i v_j^T))^2 \tag{4}$$

Simple calculation shows that $Var(v_i v_j^T) = d$, $E(v_i v_j^T) = 0$. Then $E((v_i v_j^T)^2) = d$, d is the dimension of word embedding here. Thus, $||EE^T|| \rightarrow n\sqrt{d}, n \rightarrow \infty$.

B Appendix B. Normality of RPD

Let's review the form of RPD.

$$\text{RPD}(E_1, E_2) = \frac{1}{2}\frac{||E_1E_1^T - E_2E_2^T||^2}{||E_1E_1^T||||E_2E_2^T||} \tag{5}$$

As we discuss in A, $\frac{||E_1E_1^T||||E_2E_2^T||}{n^2} \rightarrow d$, as $n \rightarrow \infty$. We only have to prove $\frac{||E_1E_1^T - E_2E_2^T||^2}{n^2}$ distributes normally. The key is how to apply the central limit theorem (CLT).

We denote as follows.

$$\begin{aligned} H_n &= \frac{||E_1E_1^T - E_2E_2^T||^2}{n^2} \\ &= \frac{||E_1E_1^T||^2 + ||E_2E_2^T||^2 - 2\langle\, E_1E_1^T, E_2E_2^T \rangle}{n^2} \end{aligned} \tag{6}$$

Notice that the term $\langle\, E_1E_1^T, E_2E_2^T \rangle$ does not contribute to the variance if we analyze the second moment of the numerator. So it is equivalent to prove $T_n = \frac{||E_1E_1^T||^2 + ||E_2E_2^T||^2}{n^2}$ distributes normally.

We project the T_n to
$S_n = \sum_{i,j}^n E(T_n|v_{ij}) - (n-1)E(T_n)$

Simple calculation would show that $\frac{Var(T_n)}{Var(S_n)} \rightarrow 1, n \rightarrow \infty, \frac{n}{d} = c$. Then by the Hajek projection theorem, we get T_n has the same distribution as S_n. It is not hard to see that each random variable $E(T_n|v_{ij})$ in S_n is independent of others. This allows us to apply CLT to S_n and get $S_n \sim \mathcal{N}(\mu, \sigma^2)$. Thus, $H_n \sim \mathcal{N}(\mu, \sigma^2)$.

C Appendix C. Monte Carlo Simulation

Here is how we perform Monte Carlo simulation. We independently produce two matrix $E_1, E_2 \in \mathbf{R}^{n\times d}$ with each entry i.i.d as $\mathcal{N}(0,1)$. Then we calculate RPD(E_1, E_2) and get the first RPD value. Repeat the process for 5000 times, we get a vector of RPDs. Drawing the histogram of this vector yields a normal distribution and we can estimate the mean and variance of the distribution by calculating the mean and variance of the vector of RPDs.

D Appendix D. Geometry Interpretation of RPD

Now we consider a general case, where $\hat{E_1}$ and $\hat{E_2}$ are embeddings with n words.

$$\begin{bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ \vdots \\ v_n^{(1)} \end{bmatrix}, \begin{bmatrix} v_1^{(2)} \\ v_2^{(2)} \\ \vdots \\ v_n^{(2)} \end{bmatrix}$$

Then

$$\begin{aligned} \frac{\langle\, \hat{E_1}, \hat{E_2} \rangle}{||\hat{E_1}||||\hat{E_2}||} &= \frac{\sum_{i=1}^n v_i^{(1)T} v_i^{(2)}}{||\hat{E_1}||||\hat{E_2}||} \\ &= \sum_{i=1}^n \frac{v_i^{(1)T} v_i^{(2)}}{||v_i^1||||v_i^{(2)}||}\frac{||v_i^1||||v_i^{(2)}||}{||\hat{E_1}||||\hat{E_2}||} \end{aligned} \tag{7}$$

We denote $\frac{||v_i^1||||v_i^{(2)}||}{||\hat{E_1}||||\hat{E_2}||}$ as w_i, $\frac{v_i^{(1)T} v_i^{(2)}}{||v_i^1||||v_i^{(2)}||}$ as $\cos(\theta_i)$

It is not hard to see that the $w_i \approx \frac{1}{n}$, when n is large enough. Then we get RPD(E_1, E_2) $\approx 1 - \frac{\sum_{i=1}^n \cos(\theta_i)}{n}$. Considering the isotropic assumption again, another observation is that the $cos(\theta_i)$ distributes normally.