Towards Lower Bounds on Number of Dimensions for Word Embeddings

Kevin Patel, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
{kevin.patel,pb}@cse.iitb.ac.in

Abstract

Word embeddings are a relatively new addition to the modern NLP researcher's toolkit. However, unlike other tools, word embeddings are used in a black box manner. There are very few studies regarding various hyperparameters. One such hyperparameter is the dimension of word embeddings. They are rather decided based on a rule of thumb: in the range 50 to 300. In this paper, we show that the dimension should instead be chosen based on corpus statistics. More specifically, we show that the number of pairwise equidistant words of the corpus vocabulary (as defined by some distance/similarity metric) gives a lower bound on the the number of dimensions, and going below this bound results in degradation of quality of learned word embeddings. Through our evaluations on standard word embedding evaluation tasks, we show that for dimensions higher than or equal to the bound, we get better results as compared to the ones below it.

1 Introduction

Word embeddings are a crucial component of modern NLP. They are learned in an unsupervised manner from large amounts of raw corpora. Bengio et al. (2003) were the first to propose neural word embeddings. Many word embedding models have been proposed since then (Collobert and Weston, 2008; Huang et al., 2012; Mikolov et al., 2013a; Levy and Goldberg, 2014).

Word vector space models can only capture differences in meaning (Sahlgren, 2006). That is, one can infer the meaning of a word by looking at its neighbors. An isolated word on its own does not mean anything in the word vector space. Thus, one needs to think of embedding algorithm's capability to capture these differences effectively, which is governed by its hyperparameters. The hyperparameters affect the information to be represented and the available degree of freedom to express it.

Most word embeddings share different design choices and hyperparameters such as context type, window size, number of dimensions of the embeddings, *etc.* However, a large portion of the research community uses word embeddings without their in-depth analysis; many proceed with default settings that come with off-the-shelf word embedding toolkits. While other hyperparameters have been studied to varying extents (see section 2), there are no rigorous studies on the number of dimensions that should be used while training word embeddings. They are usually decided via a rule of thumb (established as a side effect of other evaluations): use between 50 to 300, or by trial and error. This is a common thread across many NLP applications: Part of Speech Tagging (Collobert and Weston, 2008), Named Entity Recognition Sentence Classification (Kim, 2014), Sentiment Analysis (Liu et al., 2015), Sarcasm Detection (Joshi et al., 2016).

Depending on the corpus, its vocabulary, and the context through which the differences are elicited during training of word embedding, we are bound to obtain a certain number of words, say n, that are pairwise equidistant. Such words impose an equality constraint that the embedding algorithm has to uphold. Thus, we raise the following question:

Does n (the number of pairwise equidistant words) enforce a lower bound on the number of dimensions that should be chosen for training word embeddings on the corpus?

In this paper, we show that this seems to be true

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 31–36,
Taipei, Taiwan, November 27 – December 1, 2017

for skip gram embeddings. We show how to obtain the number of pairwise equidistant points from corpus. This number determines the lower bound. Then we show how the training algorithm of skip-gram embeddings fails to uphold the equality constraint when the number of dimensions is less than the lower bound. We show this both via analysis on toy examples as well as intrinsic evaluation on real data.

2 Background and Related Work

As mentioned earlier, the number of dimensions is often decided via the rule of thumb, or by trial and error. This holds true not only for word embedding usage but also for their evaluations.

Baroni et al. (2014) claimed that neural word embeddings are better than traditional methods such as LSA, HAL, RI (Landauer and Dumais, 1997; Lund and Burgess, 1996; Sahlgren, 2005). They experimented with different settings for the number of dimensions, but their experiments were intended to evaluate the practicality of dimensions of neural embeddings as compared to their traditional methods. However, their claim was challenged by Levy et al. (2015), who showed that **superiority of neural word embeddings is not due to the embedding algorithm, but due to certain design choices and hyperparameters optimizations**. While they investigate different hyperparameters, they keep a consistent dimension of 500 for all different embedding models that they evaluated. Many other evaluations set the number of dimensions without any justifications (Schnabel et al., 2015; Zhai et al., 2016; Ghannay et al., 2016).

Melamud et al. (2016) evaluates skip-gram word embeddings on a wide range of intrinsic and extrinsic NLP tasks. An interesting observation made by them is that while the performance for intrinsic tasks such as word pair similarity, etc. peaks at around 300 dimensions, the performance of extrinsic tasks peaked at around 50, and sometimes showed degradation for higher dimensions. This justifies the need for study of bounds for dimensions.

As is evident from the above discussion, the analysis of the number of dimensions have not received enough attention. This paper is a contribution towards that direction.

3 Motivation

Let us consider the following toy corpus of four sentences (<>is sentence separator):

<>I like cats <>I love dogs <>I hate rats <>I rate bats <>

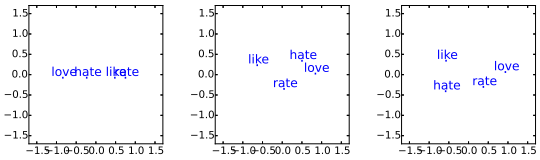
Table 1 shows the rows of the co-occurrence matrix corresponding to the four words {like, love, hate, rate}.

word	<>	I	like	love	hate	rate	rats	cats	dogs	bats
like	0	1	0	0	0	0	0	1	0	0
love	0	1	0	0	0	0	0	0	1	0
hate	0	1	0	0	0	0	1	0	0	0
rate	0	1	0	0	0	0	0	0	0	1

Table 1: Four rows corresponding to {like, love, hate, rate} of co-occurrence matrix for toy corpus

The euclidean distance between any two words from the set {like, love, hate, rate} is $\sqrt{2}$. In other words, they form a regular tetrahedron with side length $=\sqrt{2}$. The words {cats, dogs, rats, bats} form another such set. Intuitively, we know that the space which can embed a regular tetrahedron needs at least 3 dimensions. If a word embedding learning algorithm wishes to model this information correctly, it has to strive to uphold this equality constraint. However, its success will depend on the degree of freedom which it receives in terms of the number of dimensions. If it tries to embed it in a space of dimension lower than 3, then it ends up breaking the equality constraint. We end up having (0.94, 0.94), (1.77, 0.80), and (2.63, 0.10) as the average (mean, standard deviation) for the pairwise distances for dimensions 1, 2 and 3 respectively for 5 random initializations. Figure 1 shows the results of attempting to embed the regular tetrahedron created by the four words in a 1, 2, and 3-dimensional space. One can see how the algorithm fails for dimensions 1, and 2 (very high standard deviations), but succeeds in case of 3 dimensions (low standard deviation).

To further verify the distortions due to a lower than needed dimension, we make the following hypothesis: if the learning algorithm of word embeddings does not get enough dimensions, then it will fail to uphold the equality constraint. Therefore, the standard deviation of the mean of all pairwise distances will be higher. As we increase the dimension, the algorithm will get more degrees of freedom to model the equality constraint in a better way. Thus, there will be statistically significant changes in the standard deviation. Once the lower



(a) 1d(Before) (b) 2d(Before) (c) 3d(Before)

(d) 1d(After) (e) 2d(After) (f) 3d(After)

Figure 1: Trying to embed {like, love, hate, rate} in a 1,2 and 3-dimensional space. Here, *(Before)* and *(After)* indicates positions before and after training respectively. The 3 dimensional vectors in (c) and (f) are reduced to 2 dimensions using PCA for visualization purposes

bound of dimensions is reached, the algorithm gets enough degrees of freedom. Thus, from this point onwards, even if we increase dimensions, there will not be any statistically significant difference in the standard deviation.

To test this, we train word embeddings for different dimensions for an artificially created corpus with 15 pairwise equidistant words. The corpus contained sentences of the form *I* $verb_i$ $noun_i$ where $1 \leq i \leq 15$. Table 2 shows the results for the same. Note how there are statistically significant reductions (p-value < 0.05) in standard deviations up until 14 ($15 - 1$). However, once the number of dimensions is higher than 14, the differences are no longer significant (p-value > 0.05). We used Welch's Unpaired t-test for testing statistical significance.

Dim	$\overline{\sigma}$	P-value	Dim	$\overline{\sigma}$	P-value
7	0.358		12	0.154	0.0058
8	0.293	0.0020	13	0.111	0.0001
9	0.273	0.0248	14	0.044	0.0001
10	0.238	0.0313	15	0.047	0.3096
11	0.189	0.0013	16	0.054	0.1659

Table 2: Avg standard deviation ($\overline{\sigma}$) for 15 pairwise equidistant words (along with two tail p-values of Welch's unpaired t-test for statistical significance)

4 Approach

We used euclidean distance in the motivation section for ease of discussion. In practice, the metric used in conjunction with word vectors is cosine similarity. While the closed-form solution is available for the case of euclidean distance (Lower Bound = #Pairwise Equidistant points - 1) (Swanepoel, 2004), the same is not true for the case of cosine similarity. Instead, the relation between the number of dimensions and the maximum number of pairwise equiangular lines that can be embedded is an active area of research (Lemmens and Seidel, 1973; de Caen, 2000; Godsil and Roy, 2009; Barg and Yu, 2014). Table 3 gives the maximum number of pairwise equiangular lines E that can be embedded in a space of dimension λ (taken from (Barg and Yu, 2014)).

λ	E	λ	E
3	6	18	61
4	6	19	76
5	10	20	96
6	16	21	126
7<=n<=13	28	22	176
14	30	23	276
15	36	24<=n<=41	276
16	42	42	288
17	51	43	344

Table 3: Number of dimensions λ and the corresponding maximum number of equiangular lines E (for larger values of λ, refer (Barg and Yu, 2014))

To find the lower bound, one should follow the following approach:

1. Compute the word $\times$ word co-occurrence matrix from the corpus

2. Create the word $\times$ word cosine similarity matrix by treating the rows of co-occurrence matrix as word vectors

3. For each similarity value s_k:

   a) Create a graph, where the words are nodes. Create an edge between node i and node j if sim(i, j) = s_k

   b) Find maximum clique on this graph. The number of nodes in this clique is the maximum number of pairwise equidistant points E_k

   c) Reverse lookup E_k in table 3 to determine the corresponding number of dimension λ_k

4. The maximum λ among all λ_ks is the lower bound

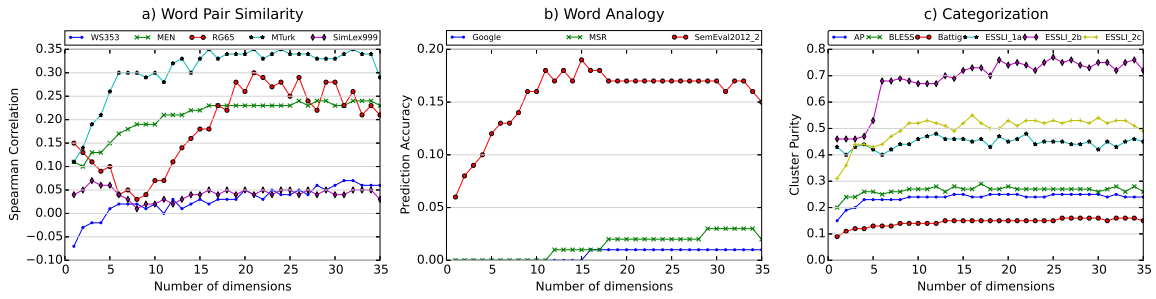


Figure 2: Performance for different tasks with respect to number of dimensions

When we applied this procedure on Brown corpus, we obtained a maximum of 62 words in step 3b), which lead to lower bound of **19 dimensions**.

A theoretical shortcoming of this approach is that finding maximum clique is NP-complete. For the Brown corpus, we obtained the maximum cliques using Parallel Maximum Clique library (PMC)(Rossi et al., 2013).

5 Experimental Setup

5.1 Word Embedding training

We train skip-gram embeddings on the Brown corpus provided with NLTK toolkit. For tokenization, we use the default tokenizer. We do not remove any stopwords. In order to control effects of randomization, we avoided it wherever possible. To this effect, we do not use negative sampling. We use hierarchical softmax to hasten the softmax computation. One word to the left and right of the input word is considered as context.

5.2 Tasks

We use the following intrinsic tasks for evaluation.

a) **Word Pair Similarity tasks** are commonly used for intrinsic evaluation of word embeddings, which involve predicting similarity between a given pair of words a and b. The evaluation involves finding cosine similarity between the embeddings of a and b, and finding the spearman correlation with human annotation. We used the WS353, MEN, RW, RG65, MTurk, and SimLex999 datasets (Faruqui and Dyer, 2014)

b) **Word Analogy tasks** are yet another commonly used tasks for intrinsic evaluation of word embeddings, which involve evaluating the accuracy of finding a missing word d in the relation: a is to b as c is to d, where (a, b) and (c, d) have the same relation. We used the Google, MSR, and SemEval 2012 Task 2 datasets (Mikolov et al., 2013b).

c) **Categorization tasks** are yet another commonly used tasks for intrinsic evaluation of word embeddings, which involve evaluating the purity of clusters formed by word embeddings. We used the AP, BLESS, ESSLI_1a, ESSLI_2b, and ESSLI_2c datasets (Schnabel et al., 2015).

6 Results and Analysis

Figure 2 shows the effects of increasing dimensions from 1 to 35 on different tasks. One observes that each series ascends till the number of dimensions reach 19, after which it stabilizes. This is because once the lower bound is reached, the errors introduced due to the violation of equality constraint are removed. Thus, the optimal performance possible with the selected configurations is reached, and the performance stabilizes thereafter.

Note that, in some cases, the performance stabilizes before 19. This is because, for that particular dataset and task, the equality constraints that are broken at lower than 19 dimensions did not matter. But, for a realistic use case, one would be better off if they stick to the lower bound.

7 Conclusion and Future Work

We discussed the importance of deciding the number of dimensions for word embedding training by looking at the corpus. We motivated the idea using abstract examples and gave an algorithm for finding the lower bound. Our experiments showed that performance of word embeddings is poor, until the lower bound is reached. Thereafter, it stabilizes. Therefore, such bounds should be used to decide the number of dimensions, instead of trial and error.

We aim to continue the work, addressing the limitations of complexity, the validity of hypothesis in extrinsic tasks, *etc.*. We will also investigate whether the same holds for different word embedding models.

Acknowledgements

We thank Arjun Atreya, Anoop Kunchukuttan, Aditya Joshi, Abhijit Mishra and other members of the Center for Indian Language Technology (CFILT) for valuable discussions and feedback.

References

Alexander Barg and Wei-Hsuan Yu. 2014. New bounds for equiangular lines. Contemporary Mathematics 625.0:111–121.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, pages 238–247.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. Journal of Machine Learning Research 3.0:1137–1155.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML). ACM, volume 307.0, pages 160–167.

Dominique de Caen. 2000. Large equiangular sets of lines in euclidean space. Electronic Journal of Combinatorics 7.0.

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

Sahar Ghannay, Benoit Favre, Yannick Estve, and Nathalie Camelin. 2016. Word embedding evaluation and combination. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC). European Language Resources Association (ELRA).

Chris Godsil and Aidan Roy. 2009. Equiangular lines, mutually unbiased bases, and spin models. European Journal of Combinatorics 30.0:246–262.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, pages 873–882.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 1006–1011.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pages 1746–1751.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review 104.0:211–240.

Petrus WH Lemmens and Johan J Seidel. 1973. Equiangular lines. Journal of Algebra 24.0:494–512.

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

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association of Computational Linguistics 3.0:211–225.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 1433–1443.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments, & Computers 28.0:203–208.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, pages 1030–1040.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.. pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, pages 746–751.

Ryan A Rossi, David F Gleich, Assefaw H Gebremedhin, Mostofa A Patwary, Ryan A Rossi, David F Gleich, David F Gleich, and Ryan A Rossi. 2013. A fast parallel maximum clique algorithm for large sparse graphs and temporal strong components. arXiv preprint 1302.6256 .

Magnus Sahlgren. 2005. An introduction to random indexing. In In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, (TKE).

Magnus Sahlgren. 2006. The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces. Ph.D. thesis, Stockholm University.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 298–307.

Konrad J Swanepoel. 2004. Equilateral sets in finite-dimensional normed spaces. In Seminar of Mathematical Analysis. volume 71.0, pages 195–237.

Michael Zhai, Johnny Tan, and Jinho D. Choi. 2016. Intrinsic and extrinsic evaluations of word embeddings. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, pages 4282–4283.