# Stability of Debiased Word Embeddings

Alejandro Pelcastre
University of California, Berkeley
alejandro.pelcas@berkeley.edu

April 11, 2022

**Abstract**

Word embeddings are unstable and can lead to volatile machine learning models. In this paper, I attempt to mitigate the unstable nature of embedding spaces by debiasing the word embeddings. The baseline average stability for 300 words was around 0.38% while the average stability for the same words in the debiased embedding spaces were around 0.7%. It should be noted that these differences in stability may have been due to the difference in embedding methods rather than the debiasing process. Mainly, I find that the debiasing methods for word embeddings by Bolukbasi [1] is a mostly stable process as it retains most of the words nearest neighbors.

## 1 Introduction

Machine Learning has become a ubiquitous tool in industry, academia, and beyond to solve complex problems. The abundance of data generated every day continues to fuel the power and capabilities of machine learning. Often one uses machine learning to make predictions about the future, however, there is an instability problem in machine learning when we consider word embeddings.

In order to stay up-to-date, many machine learning models need to be retrained to take advantage of new streaming data or simply to improve with more data [2]. There is, however, a minor issue. Machine retraining can lead to widely different results when training on word embeddings. This makes machine learning models sensitive to changes.

Word embeddings are low dimensional word vectors $\vec{w} \in R^d$ that is a powerful tool used in several Natural Language Processing (NLP) tasks. They are easily trained and work well for transfer learning. These embeddings themselves, however, are a source of instability for machine learning models. Retraining on different corpora, different data sizes, or different algorithms can lead to significantly different representations of each word. This can lead to further problems on downstream tasks that rely on word embeddings and could leave results that do not account for this instability to have invalid conclusions. While there has been research into identifying and mitigating the instability of word embeddings, there is seldom if any research to my knowledge of word embedding stability in debiased data sets.

## 2 Background

Until recently, most researchers did not focus on the implications of a biased dataset [1]. Data sets that do not account for the stereotypes and biases that plague our texts and literature will perpetuate them in the machine learning models. An example of how these biases can plague a model is when we consider word similarities using word embeddings of gendered words. Consider the prominent example of these similar word distances:

$$m\vec{a}n - wo\vec{m}an \approx ki\vec{n}g - qu\vec{e}en$$

These are examples of acceptable *gender specific* word similarities - words with established gender. However, an example of inappropriate word similarities occurs when we consider results like the man - woman ≈ computer programmer - homemaker:

Debiasing a dataset means that relationships like the one above will not appear when training the word embeddings. Since word relationships are more likely to be neutral after debiasing there is a possibility of leading them to be more stable.

In this paper I will be using pre-trained debiased word embeddings from Tolga, Kalai, and Iniesta's repo (https://github.com/tolga-b/debiaswe) which are trained on the google news dataset using word2vec, as well as the regular google news embeddings all with embedding dimension 300. To debias the word embeddings, they used the **hard de-biasing** process [1].

Previous work [3] suggests that we should not rely on a single word embedding to evaluate a change of distances so we will use GloVe embeddings trained on the wiki-giga data as well. (you can download after importing gensim.downloader and running gensim.downloader.load('glove-wiki-gigaword-300')).

## 3 Methods

Word Embeddings are low-dimensional vector representation of vocabulary words that contain semantic and relational information of other similar words. There are several types of Word Embedding Methods (WEMs) such as Word2Vec, GloVe, fastText, TF-IDF, Bert embeddings, and many more. Given a corpus $C_i$ and different WEMs $W_{j,k}$ where the subscript $j, k$ indicates that there are different WEMs $j$ with different parameters $k$ at play (for example, the size of the dimension), we can generate an embedding space $E_{C_i,W_{j,k}}$ which we will condense to the following notation: $E_{ijk}$.

Given a word $w \in C$, the embedding it takes on in embedding space will be denoted as $E_{ijk}(w)$. While there are several evaluation methods to formalize the notion of stability, N. Griffis [6] concludes that nearest neighbor information alone is sufficient to capture most of the performance benefits derived from using pre-trained word embeddings. Thus, we get to our definition of stability: Given a word $w \in C$ and two embedding spaces $E_{ijk}, E_{i'j'k'}$, the *stability* of a word $w$ is given as the overlap between nearest neighbors in the two embedding spaces:

$$S(w) = \frac{|KNN(\vec{w})_{E_{ijk}} \cap KNN(\vec{w})_{E_{i'j'k'}}|}{k}$$

where $k$ is the number of k-nearest neighbors. We will use k=10 for the rest of the paper as ten is found in various other NLP tasks [7,8] as serves as a reputable metric number.

The 10 nearest neighbors will be computed with the popular word embedding distance metric, *cosine similarities* as it will make our results more consistent and compatible with other research even though other metrics like $l^1$ norm, $l^2$ norm and other give similar results [4].

## 4 Results and Discussion

We perform a regular stability check on a list of 300 words using the GloVe embeddings from the. glove-wiki-giga dataset and the Word2Vec embeddings from the google-news dataset. We then feed these two embedding spaces and each word in our 300 word set to get the stabilities of the words in these spaces. We get the following plot where the number of words is in multiples of 10.

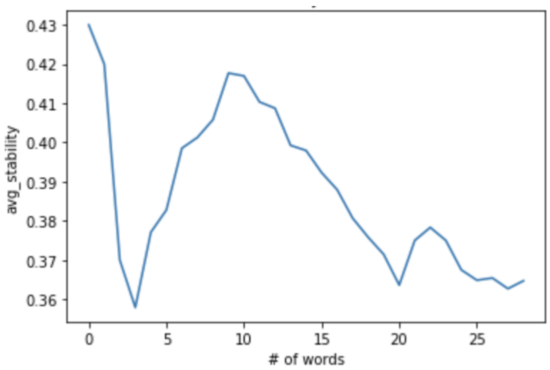

Figure 1: On the x-axis we have the size of the number of words in 10s that we used to evaluate the stabilities. On the y-axis we have the avg stability of the words used

In figure 1 we see how the average stability of the words zigzag around the .4 mark. This behavior may be due to the specific words chosen to evaluate their stability. Then we take the same approach but this time apply it to the debiased google news embeddings. Similarly, we find the average 10 nearest neighbors that intersect with the word2vec google news embeddings and see how the stable the same 300 words perform:

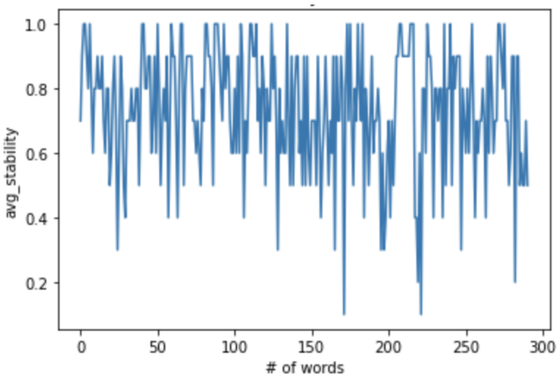

Figure 2: On the x-axis we have the size of the number of words that we used to evaluate the stabilities of the debiased embeddings. On the y-axis we have the stability of the words used

In figure 2 we have the actual stabilities on the y-axis and the amount of words on the x-axis. Notice how these also zigzag, but at much elevated baseline. We do not expect perfect stability across the two embeddings even though they are similar. This is because the debiasing is meant to move gendered words in a way to equalize their distance from non-gendered words like professions or sports etc.

Using four different embedding spaces we find that the debiased embedding and the regular embedding on the google news corpus produce around 0.7% stability average per word. The other two embedding spaces produced around 0.38% stability accuracy. While these differences seem significant, it is not all explained by the usefulness of the debiased process. The main result is that the embeddings are mostly stable after debiasing them since they retain almost 8 of the same nearest neighbor words in the embedding spaces on average.

## 4.1 Limitations

There were notable hurdles when progressing through this project. First, I quickly used up my computers memory when working with the embedding datasets and the virtual machines were unreliable since they would disconnect frequently meaning I needed to reboot all the necessary data. This led me to use unideal embedding spaces such the Word2Vec and GloVe methods on different corpuses for the biased experiemnt (see figure 1). Additionally, parsing the data alone was computationally expensive and I could not implement my own debiasing method. Instead, I used ones already trained (see https://github.com/tolga-b/debiaswe). I learned a lot along this journey and perhaps made the mistake of taking on this task alone. In the end, I learned first-hand about the life-cycle of the data process and realized why many data scientists claim cleaning data is the most time consuming (it was). Finally, while I managed to produce the graphs in the report, I was not able to change or again produce more plots due to some unknown problem with my computer running out of memory.

## 5 Conclusion

In essence we were able to show that debiasing word embeddings leaves most of the nearest neighbors unchanged. The stability of debiasing is nearly twice as much as those of comparing two embedd Future work has several methods for improving this project. First, being able to compare more words than 300 to check the avg stability can be greatly increased with more resources. Second, different embedding methods should be approached such as fastText as well as with other datasets. Ideally there would be an ensemble of embedding methods [3] on different corpuses compared against the same methods on debiased corpuses. One would need access to several resoures in order to perform this process on larger sets and word counts. Third, the difference in stability between the non-debiased and the biased embeddings would be more impactful if for the K nearest neighbors algorithm you compare the embeddings using the same method (i.e. word2vec or GloVe) on a corpus C and its debiased version. Fourth, as other researchers suggest [4] we should focus on the stability of medium-frequency words or words that have a frequency of around 2 to 500. Higher frequencies than these ten to produce more stable embeddings, but these middle ones have a high variance. When conducting my stability measures I did not do so with any particular attention to the frequencies of the words. A more careful analysis would take this numbers into account. Fifth, playing with the parameters could also yield interesting results such as changing the dimension of the embeddings or checking the stabilities of different languages to see if there are linguistic characteristics that impact the stability.

## References

[1] T. Bolukbasi, K. Chang, J. Zou, Venkatesh Saligrama, Adam Kalain
Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, arxiv, 1607.06520., 2016

[2] M. Leszczynski, A. May, J. Zhang, S. Wu, C. R. Aberger, C. Re
Understanding The Downstream Instability of Word Embeddings, arxiv, 2003.04983v1

[3] M. Antoniak, D. Mimno
Evaluating the Stability of Embedding-based Word Similarities, 2018

[4] L. Wendalndt, J.K. Kummerfeld, R. Mihalcea
Factors Influencing the Surprising Instability of Word Embeddings, 2018

[5] A. Borah, M. P. Barman, A. Awekar
Are Word Embedding Methods Stable and Should We Care About It?, 2021

[6] N. Griffis, F. Lussier
Second-Order Word Embeddings from Nearest Neighbor Topological Features, 2017

[7] D. McCarthy, R. Navigli
SemEval-2007 Task 10: English Lexical Substitution Task, 2007

[8] A. Garimella, C. Banea, R.Mihalcea
Demographic-Aware Word Associations, 2017