Word Vectors from Small Corpora

Eric Zelikman
Stanford University
ezelikman@stanford.edu

Abstract

Word vectors have become a standard tool for interpreting relationships between words, also used extensively in other learning tasks. As indicated in Altszyler et al. (2016), typical models use datasets on order of millions of tokens to produce especially meaningful results. This paper presents a new approach to generating word vector representations which results in performance comparable to more extensively trained word vectors with less corpus data. Building off of recent work relating the statistics of a word vector distribution to effective sentence vectors (Zelikman, 2018), this model generates word vectors based on the assumption that words take on roles to assign the largest possible amount of meaning to the sentences that they appear in.

1 Introduction

Word vectors allow us to better understand and model a language and its connotations, but traditionally, word vectors rely on the ability to gather inhuman amounts of data in a way that weakens their applicability to less documented languages or encourages a transfer learning approach for smaller communities. Although typical word vectors techniques can be applied to small corpora, looking directly at co-occurrences or otherwise local analysis ignores the role that a word plays in a sentence.

The basic assumption of this paper is that, although more surprising word meanings play a larger role in defining a sentence, the meaning of a word should be as close to the meanings of the sentences in which it appears as possible. In an inherently cyclic way, the words in a sentence both define its meaning and are partially defined by the sentence. Using a formula for word representations proposed in Zelikman (2018), this paper shows that modifying word vectors (and implicitly, their calculated sentence vectors) allows for useful results.

This paper also compares different approaches to implementing this general idea, notably highlighting the difference in effectiveness between an attempt to generate initial meanings constructively or randomly and the performance of different iteration approaches after that.

2 Related Work

There are a number of ways currently to generate word vectors. Most prominently, in chronological order of development: word2vec is fairly well known and a precursor to most other models, using a neural network to attempt to predict a word from its surroundings (Mikolov et al., 2013). GloVe instead uses co-occurrences and makes a lower dimensional representation of the co-occurrences matrix that loses as little information as possible (Pennington et al., 2014). fastText uses a set of other optimizations over word2vec, but primarily the use of character-level prediction of word meaning which allows for out-of-context words (Joulin et al., 2016).

3 Dataset and Features

Two datasets were used: for qualitative analysis and as an example of a real small corpus, approximately 2,000 sentences of a reddit 2009 comments corpus was used. This proved to be an interesting dataset due to the various idiosyncrasies associated with the particular communities included. For the quantitative comparison, it seemed necessary to include a dataset which could be used as a universal benchmark. Choosing a fairly diverse but larger (20,000 sentences) dataset, Stanford's Sentiment Treebank was used.

Table 1: Closest elements to cluster centers generated by the k-means clustering on the model-generated word vectors from a small (2000 sentence). Additionally, the right-most column is a cluster from word vectors generated on the SST. Note that the typo "becuase" is, in fact, part of the corpus.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	SST Cluster
starts	solution	complaint	imprison	punchier
becuase	beneficial	gcc	inbuilt	crisper
origins	puppies	vud	burglar	modernizes
postings	hubble	dispatch	traditionally	hankies
ever	valid	conf	exhausted	skidding
interested	proved	requiring	undeserving	repugnant
intact	efforts	rebuild	industrialized	feathers
steel	worth	upstream	tradition	decision

4 Methods

4.1 Word Vector Generation

In order to generate the word vectors, the following technique was used (Note that sentence vectors were calculated using the algorithm in Zelikman (2018), which, roughly speaking, weights each word's embedding by its unexpectedness in the context of the sentence, measured with a Mahalanobis-distance based approach):

1. Randomly sort vocabulary, with selection probability being frequency in dataset
2. For each word:

   Initiate a unit vector randomly (normalized Gaussian)

   Calculate sentence vectors of all the sentences where the word appears, with only initiated words

   Bring the word closer to the normalized average of the sentence vectors
3. Proceed by selecting random words or sentences and adjusting the word vectors in each to be as close as possible to their corresponding sentence vectors (Parallelized)

4.2 Word Vector Generation

5 Evaluation

5.1 Qualitative

As shown in the Table 1, the clusters generated even for fairly small datasets seem quite unique. Unfortunately, as indicated by the SST results, the exact manner in which these word vectors are related seems somewhat unclear. It seems that, broadly, this representation is more reflective of a kind of relatedness rather than similarity, a distinction discussed in Ballatore et al. (2014). It is unclear whether the dimensions themselves represent anything explicit. For example "king" - "man" + "woman" using the Stanford Sentiment Treebank vectors (Weighted by their unexpectednesses in the overall corpus) returns "renaissance" and "spain" as the closest vectors*.

One of the most interesting results is that, when taking the constructive approach, words tend to end up related to the same words, which is encouraging. Curiously, sometimes, as in the second column of Table 2, ("puppies") sufficiently unrelated concepts seem to be able to coexist in this representation. The most encouraging justification for this property is that, although the overall algorithm is normalized by the use of the Mahalanobis metric, there is nothing that inherently penalizes sufficiently different categories overlapping unless the dataset grows large. In fact, it is efficient to pack maximally unrelated phrases together if they happen to be learned alongside one another and are different from their surrounding vectors in a consistent way. This is arguably a weakness of this approach, but is somewhat reflective of learning: it is† not especially uncommon to connect unrelated concepts because they were learned in the same context.

5.2 Quantitative

Of course, one interesting question is, if a particular dimension along these vectors doesn't appear to directly map to some particular dimension of meaning in the language, is it useful for the purpose of an algorithm? so, it seems necessarily to evaluate these vectors beyond just looking at groups of them and making (perhaps tenuous) connections. Furthermore, whatever quantitative task is performed should be at least partially order-dependent, since the training mechanism of the word vectors doesn't explicitly take into account word order. For this reason, Stanford's Sentiment Treebank was chosen as the task, and a simple publicly available pytorch implementation of a BiLSTM for this task was used as a baseline, with some basic hyperparameter tuning performed (Tian).

Unfortunately, a BiLSTM is a fairly powerful tool in the first place. However, as shown in the

*Alongside "woman" itself...

†From personal experience, at least

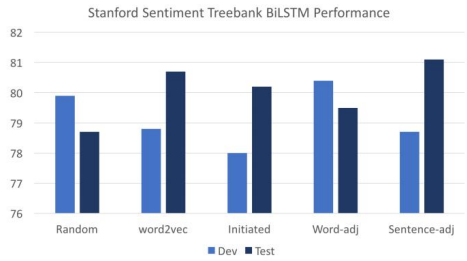

Figure 1: A comparison of the dev and test performances (accuracies) of different kinds of vectors

following table, the performance of the initiated and then adjusted by sentence vectors was highest.

6 Discussion

This comes with a number of interesting theoretical strengths that will be explored: First, words that are more common will need to be closer to more sentences, and thus will develop a less distinct meaning. Ideally, words that are rare should not develop a completely random meaning, but one that both incorporates the other words in the sentences where it appears, but also makes the assumption that it provides meaning of its own.

7 Conclusion and Future Work

By far the most valuable realization that comes from this research, I believe, is the way in which it suggests a reframing of conceptualization and construction of mental representations: it appears that the process of representing the meaning of a collection of things can meaningfully be used to help define each of them.

Acknowledgments

A thank you to Abhijeet Shenoi, my mentor for this project.

References

Edgar Altszyler, Mariano Sigman, Sidarta Ribeiro, and Diego Fernández Slezak. 2016. Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. *ArXiv e-prints*, page arXiv:1610.01520.

A. Ballatore, M. Bertolotto, and D. C. Wilson. 2014. An evaluative baseline for geo-semantic relatedness and similarity. *ArXiv e-prints*.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *ArXiv e-prints*.

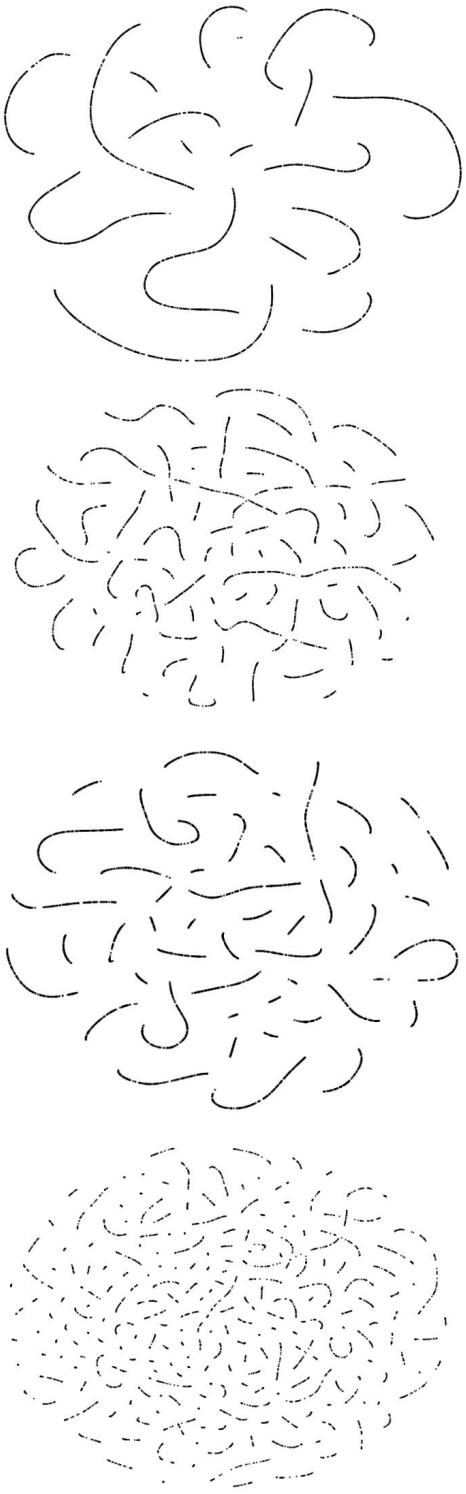

Figure 2: The general vectorial edit distance model. Given two input words, it first extracts the phonemes, then finds the lowest-vector-cost set of substitutions and insertions/deletions

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Claire Tian. Pytorch Sentiment Classification.

Eric Zelikman. 2018. Context is everything: Finding meaning statistically in semantic spaces. *CoRR*, abs/1803.08493.