Evaluating the Stability of Embedding-based Word Similarities

Maria Antoniak
Cornell University
maa343@cornell.edu

David Mimno
Cornell University
mimno@cornell.edu

Abstract

Word embeddings are increasingly being used as a tool to study word associations in specific corpora. However, it is unclear whether such embeddings reflect enduring properties of language or if they are sensitive to inconsequential variations in the source documents. We find that nearest-neighbor distances are highly sensitive to small changes in the training corpus for a variety of algorithms. For all methods, including specific documents in the training set can result in substantial variations. We show that these effects are more prominent for smaller training corpora. We recommend that users never rely on single embedding models for distance calculations, but rather average over multiple bootstrap samples, especially for small corpora.

1 Introduction

Word embeddings are a popular technique in natural language processing (NLP) in which the words in a vocabulary are mapped to low-dimensional vectors. Embedding models are easily trained—several implementations are publicly available—and relationships between the embedding vectors, often measured via cosine similarity, can be used to reveal latent semantic relationships between pairs of words. Word embeddings are increasingly being used by researchers in unexpected ways and have become popular in fields such as digital humanities and computational social science (Hamilton et al., 2016; Heuser, 2016; Phillips et al., 2017).

Embedding-based analyses of semantic similarity can be a robust and valuable tool, but we find that standard methods dramatically under-represent the variability of these measurements. Embedding algorithms are much more sensitive than they appear to factors such as the presence of specific documents, the size of the documents, the size of the corpus, and even seeds for random number generators. *If users do not account for this variability, their conclusions are likely to be invalid.* Fortunately, we also find that simply averaging over multiple bootstrap samples is sufficient to produce stable, reliable results in all cases tested.

NLP research in word embeddings has so far focused on a *downstream-centered* use case, where the end goal is not the embeddings themselves but performance on a more complicated task. For example, word embeddings are often used as the bottom layer in neural network architectures for NLP (Bengio et al., 2003; Goldberg, 2017). The embeddings' training corpus, which is selected to be as large as possible, is only of interest insofar as it generalizes to the downstream training corpus.

In contrast, other researchers take a *corpus-centered* approach and use relationships between embeddings as direct evidence about the language and culture of the authors of a training corpus (Bolukbasi et al., 2016; Hamilton et al., 2016; Heuser, 2016). Embeddings are used as if they were simulations of a survey asking subjects to free-associate words from query terms. Unlike the downstream-centered approach, the corpus-centered approach is based on direct human analysis of nearest neighbors to embedding vectors, and the training corpus is not simply an off-the-shelf convenience but rather the central object of study.

Transactions of the Association for Computational Linguistics, vol. 6, pp. 107–119, 2018. Action Editor: Ivan Titov.
Submission batch: 6/2017; Revision batch: 9/2017; Published 2/2018.

Downstream-centered	Corpus-centered
Big corpus	Small corpus, difficult or impossible to expand
Source is not important	Source is the object of study
Only vectors are important	Specific, fine-grained comparisons are important
Embeddings are used in downstream tasks	Embeddings are used to learn about the mental model of word association for the authors of the corpus

Table 1: Comparison of *downstream-centered* and *corpus-centered* approaches to word embeddings.

While word embeddings may appear to measure properties of language, they in fact only measure properties of a curated corpus, which could suffer from several problems. The training corpus is merely a *sample* of the authors' language model (Shazeer et al., 2016). Sources could be missing or over-represented, typos and other lexical variations could be present, and, as noted by Goodfellow et al. (2016), "Many datasets are most naturally arranged in a way where successive examples are highly correlated." Furthermore, embeddings can vary considerably across random initializations, making lists of "most similar words" unstable.

We hypothesize that training on small and potentially idiosyncratic corpora can exacerbate these problems and lead to highly variable estimates of word similarity. Such small corpora are common in digital humanities and computational social science, and it is often impossible to mitigate these problems simply by expanding the corpus. For example, we cannot create more 18th Century English books or change their topical focus.

We explore causes of this variability, which range from the fundamental stochastic nature of certain algorithms to more troubling sensitivities to properties of the corpus, such as the presence or absence of specific documents. We focus on the training corpus as a source of variation, viewing it as a *fragile* artifact curated by often arbitrary decisions. We examine four different algorithms and six datasets, and we manipulate the corpus by shuffling the order of the documents and taking bootstrap samples of the documents. Finally, we examine the effects of these manipulations on the cosine similarities between embeddings.

We find that there is considerable variability in embeddings that may not be obvious to users of these methods. Rankings of most similar words are not reliable, and both ordering and membership in such lists are liable to change significantly. Some uncertainty is expected, and there is no clear criterion for "acceptable" levels of variance, but we argue that the amount of variation we observe is sufficient to call the whole method into question. For example, we find cases in which there is zero set overlap in "top 10" lists for the same query word across bootstrap samples. Smaller corpora and larger document sizes increase this variation. Our goal is to provide methods to quantify this variability, and to account for this variability, we recommend that as the size of a corpus gets smaller, cosine similarities should be averaged over many bootstrap samples.

2 Related Work

Word embeddings are mappings of words to points in a K-dimensional continuous space, where K is much smaller than the size of the vocabulary. Reducing the number of dimensions has two benefits: first, large, sparse vectors are transformed into small, dense vectors; and second, the conflation of features uncovers latent semantic relationships between the words. These semantic relationships are usually measured via cosine similarity, though other metrics such as Euclidean distance and the Dice coefficient are possible (Turney and Pantel, 2010). We focus on four of the most popular training algorithms: Latent Semantic Analysis (LSA) (Deerwester et al., 1990), Skip-Gram with Negative Sampling (SGNS) (Mikolov et al., 2013), Global Vectors for Word Representation (GloVe) (Pennington et al., 2014), and Positive Pointwise Mutual Information (PPMI) (Levy and Goldberg, 2014) (see Section 5 for more detailed descriptions of these algorithms).

In NLP, word embeddings are often used as features for downstream tasks. Dependency parsing (Chen and Manning, 2014), named entity recognition (Turian et al., 2010; Cherry and Guo, 2015), and bilingual lexicon induction (Vulic and Moens, 2015) are just a few examples where the use of embeddings as features has increased performance in recent years.

Increasingly, word embeddings have been used as evidence in studies of language and culture. For example, Hamilton et al. (2016) train separate embeddings on temporal segments of a corpus and then

analyze changes in the similarity of words to measure semantic shifts, and Heuser (2016) uses embeddings to characterize discourse about virtues in 18th Century English text. Other studies use cosine similarities between embeddings to measure the variation of language across geographical areas (Kulkarni et al., 2016; Phillips et al., 2017) and time (Kim et al., 2014). Each of these studies seeks to reconstruct the mental model of authors based on documents.

An example that highlights the contrast between the downstream-centered and corpus-centered perspectives is the exploration of implicit bias in word embeddings. Researchers have observed that embedding-based word similarities reflect cultural stereotypes, such as associations between occupations and genders (Bolukbasi et al., 2016). From a downstream-centered perspective, these stereotypical associations represent bias that should be filtered out before using the embeddings as features. In contrast, from a corpus-centered perspective, implicit bias in embeddings is not a problem that must be fixed but rather a means of measurement, providing quantitative evidence of bias in the training corpus.

Embeddings are usually evaluated on direct use cases, such as word similarity and analogy tasks via cosine similarities (Mikolov et al., 2013; Pennington et al., 2014; Levy et al., 2015; Shazeer et al., 2016). Intrinsic evaluations like word similarities measure the *interpretability* of the embeddings rather than their downstream task performance (Gladkova and Drozd, 2016), but while some research does evaluate embedding vectors on their downstream task performance (Pennington et al., 2014; Faruqui et al., 2015), the standard benchmarks remain intrinsic.

There has been some recent work in evaluating the stability of word embeddings. Levy et al. (2015) focus on the hyperparameter settings for each algorithm and show that hyperparameters such as the size of the context window, the number of negative samples, and the level of context distribution smoothing can affect the performance of embeddings on similarity and analogy tasks. Hellrich and Hahn (2016) examine the effects of word frequency, word ambiguity, and the number of training epochs on the reliability of embeddings produced by the SGNS and skip-gram hierarchical softmax (SGHS) (a variant of SGNS), striving for reproducibility and recommending against sampling the corpus in order to preserve stability. Likewise, Tian et al. (2016) explore the robustness of SGNS and GloVe embeddings trained on large, generic corpora (Wikipedia and news data) and propose methods to align these embeddings across different iterations.

In contrast, our goal is not to produce artificially stable embeddings but to identify the factors that create instability and measure our statistical confidence in the cosine similarities between embeddings trained on small, specific corpora. We focus on the corpus as a fragile artifact and source of variation, considering the corpus itself as merely a *sample* of possible documents produced by the authors. We examine whether the embeddings accurately model those authors, using bootstrap sampling to measure the effects of adding or removing documents from the training corpus.

3 Corpora

We collected two sub-corpora from each of three datasets (see Table 2) to explore how word embeddings are affected by size, vocabulary, and other parameters of the training corpus. In order to better model realistic examples of corpus-centered research, these corpora are deliberately chosen to be publicly available, suggestive of social research questions, varied in corpus parameters (e.g. topic, size, vocabulary), and much smaller than the standard corpora typically used in training word embeddings (e.g. Wikipedia, Gigaword). Each dataset was created organically, over specific time periods, in specific social settings, by specific authors. Thus, it is impossible to expand these datasets without compromising this specificity.

We process each corpus by lowercasing all text, removing words that appear fewer than 20 times in the corpus, and removing all numbers and punctuation. Because our methods rely on bootstrap sampling (see Section 6), which operates by removing or multiplying the presence of documents, we also remove duplicate documents from each corpus.

U.S. Federal Courts of Appeals The U.S. Federal courts of appeals are regional courts that decide appeals from the district courts within their federal judicial circuit. We examine the embeddings of the most recent five years of the 4th and 9th circuits.[1]

[1] https://www.courtlistener.com/

Corpus	Number of documents	Unique words	Vocabulary density	Words per document
NYT Sports (2000)	8,786	12,475	0.0020	708
NYT Music (2000)	3,666	9,762	0.0037	715
AskScience	331,635	16,901	0.0012	44
AskHistorians	63,578	9,384	0.0022	66
4th Circuit	5,368	16,639	0.0014	2,281
9th Circuit	9,729	22,146	0.0011	2,108

Table 2: Comparison of the number of documents, number of unique words (after removing words that appear fewer than 20 times), vocabulary density (the ratio of unique words to the total number of words), and the average number of words per document for each corpus.

Setting	Method	Tests...	Run 1	Run 2	Run 3
Fixed	Documents in consistent order	variability due to algorithm (baseline)	A B C	A B C	A B C
Shuffled	Documents in random order	variability due to document order	A C B	B A C	C B A
Bootstrap	Documents sampled with replacement	variability due to document presence	B A A	C A B	B B B

Table 3: The three settings that manipulate the document order and presence in each corpus.

The 4th circuit contains Washington D.C. and surrounding states, while the 9th circuit contains the entirety of the west coast. Social science research questions might involve measuring a widely held belief that certain courts have distinct ideological tendencies (Broscheid, 2011). Such bias may result in measurable differences in word association due to framing effects (Card et al., 2015), which could be observable by comparing the words associated with a given query term. We treat each opinion as a single document.

New York Times The New York Times (NYT) Annotated Corpus (Sandhaus, 2008) contains newspaper articles tagged with additional metadata reflecting their content and publication context. To constrain the size of the corpora and to enhance their specificity, we extract data only for the year 2000 and focus on only two sections of the NYT dataset: sports and music. In the resulting corpora, the sports section is substantially larger than the music section (see Table 2). We treat an article as a single document.

Reddit Reddit[2] is a social website containing thousands of forums (subreddits) organized by topic. We use a dataset containing all posts for the years 2007-2014 from two subreddits: `/r/AskScience` and `/r/AskHistorians`. These two subreddits allow users to post any question in the topics of history and science, respectively. *AskScience* is more than five times larger than *AskHistorians*, though the document length is generally longer for *AskHistorians* (see Table 2). Reddit is a popular data source for computational social science research; for example, subreddits can be used to explore the distinctiveness and dynamicity of communities (Zhang et al., 2017). We treat an original post as a single document.

[2] `https://www.reddit.com/`

4 Corpus Parameters

Order and presence of documents We use three different methods to sample the corpus: FIXED, SHUFFLED, and BOOTSTRAP. The FIXED setting includes each document exactly once, and the documents appear in a constant, chronological order across all models. The purpose of this setting is to measure the baseline variability of an algorithm, independent of any change in input data. Algorithmic variability may arise from random initializations of learned parameters, random negative sampling, or randomized subsampling of tokens within documents. The SHUFFLED setting includes each document exactly once, but the order of the documents is randomized for each model. The purpose of this setting is to evaluate the impact of variation on how we present examples to each algorithm. The order of documents could be an important factor for algorithms that use online training such as SGNS. The BOOTSTRAP setting samples N documents randomly with replacement, where N is equal to the number of documents in the FIXED setting. The purpose of this setting is to measure how much variability is due to the presence or absence of specific sequences of

tokens in the corpus. See Table 3 for a comparison of these three settings.

Size of corpus We expect the stability of embedding-based word similarities to be influenced by the size of the training corpus. As we add more documents, the impact of any specific document should be less significant. At the same time, larger corpora may also tend to be more broad in scope and variable in style and topic, leading to less idiosyncratic patterns in word co-occurrence. Therefore, for each corpus, we curate a smaller sub-corpus that contains 20% of the total corpus documents. These samples are selected using contiguous sequences of documents at the beginning of each training (this ensures that the FIXED setting remains constant).

Length of documents We use two document segmentation strategies. In the first setting, each training instance is a single *document* (i.e. an article for the NYT corpus, an opinion from the Courts corpus, and a post from the Reddit corpus). In the second setting, each training instance is a single *sentence*. We expect this choice of segmentation to have the largest impact on the BOOTSTRAP setting. Documents are often characterized by "bursty" words that are locally frequent but globally rare (Madsen et al., 2005), such as the name of a defendant in a court case. Sampling whole documents with replacement should magnify the effect of bursty words: a rare but locally frequent word will either occur in a Bootstrap corpus or not occur. Sampling sentences with replacement should have less effect on bursty words, since the chance that an entire document will be removed from the corpus is much smaller.

5 Algorithms

Evaluating all current embedding algorithms and implementations is beyond the scope of this work, so we select four categories of algorithms that represent distinct optimization strategies. Recall that our goal is to examine how algorithms respond to variation in the *corpus*, not to maximize performance in the accuracy or effectiveness of the embeddings.

The first category is online stochastic updates, in which the algorithm updates model parameters using stochastic gradients as it proceeds through the training corpus. All methods implemented in the `word2vec` and `fastText` packages follow this format, including skip-gram, CBOW, negative sampling, and hierarchical softmax (Mikolov et al., 2013). We focus on SGNS as a popular and representative example. The second category is batch stochastic updates, in which the algorithm first collects a matrix of summary statistics derived from a pass through the training data that takes place before any parameters are set, and then updates model parameters using stochastic optimization. We select the GloVe algorithm (Pennington et al., 2014) as a representative example. The third category is matrix factorization, in which the algorithm makes deterministic updates to model parameters based on a matrix of summary statistics. As a representative example we include PPMI (Levy and Goldberg, 2014). Finally, to test whether word order is a significant factor we include a document-based embedding method that uses matrix factorization, LSA (Deerwester et al., 1990; Landauer and Dumais, 1997).

These algorithms each include several hyperparameters, which are known to have measurable effects on the resulting embeddings (Levy et al., 2015). We have attempted to choose settings of these parameters that are commonly used and comparable across algorithms, but we emphasize that a full evaluation of the effect of each algorithmic parameter would be beyond the scope of this work. For each of the following algorithms, we set the context window size to 5 and the embeddings size to 100. Since we remove words that occur fewer than 20 times during preprocessing of the corpus, we set the frequency threshold for the following algorithms to 0.

For all other hyperparameters, we follow the default or most popular settings for each algorithm, as described in the following sections.

5.1 LSA

Latent semantic analysis (LSA) factorizes a sparse term-document matrix X (Deerwester et al., 1990; Landauer and Dumais, 1997). X is factored using singular value decomposition (SVD), retaining K singular values such that

$$X \approx X_K = U_K \Sigma_K V_K^T.$$

The elements of the term-document matrix are weighted, often with TF-IDF, which measures the

importance of a word to a document in a corpus. The dense, low-rank approximation of the term-document matrix, X_K, can be used to measure the relatedness of terms by calculating the cosine similarity of the relevant rows of the reduced matrix.

We use the `sci-kit learn`[3] package to train our LSA embeddings. We create a term-document matrix with TF-IDF weighting, using the default settings except that we add L2 normalization and sublinear TF scaling, which scales the importance of terms with high frequency within a document. We perform dimensionality reduction via a randomized solver (Halko et al., September 2009).

The construction of the term-count matrix and the TF-IDF weighting should introduce no variation to the final word embeddings. However, we expect variation due to the randomized SVD solver, even when all other parameters (training document order, presence, size, etc.) are constant.

5.2 SGNS

The skip-gram with negative sampling (SGNS) algorithm (Mikolov et al., 2013) is an online algorithm that uses randomized updates to predict words based on their context. In each iteration, the algorithm proceeds through the original documents and, at each word token, updates model parameters based on gradients calculated from the current model parameters. This process maximizes the likelihood of observed word-context pairs and minimizes the likelihood of negative samples.

We use an implementation of the SGNS algorithm included in the Python library `gensim`[4] (Řehůřek and Sojka, 2010). We use the default settings provided with `gensim` except as described above.

We predict that multiple runs of SGNS on the same corpus will not produce the same results. SGNS randomly initializes all the embeddings before training begins, and it relies on negative samples created by randomly selecting word and context pairs (Mikolov et al., 2013; Levy et al., 2015). We also expect SGNS to be sensitive to the order of documents, as it relies on stochastic gradient descent which can be biased to be more influenced by initial documents (Bottou, 2012).

[3] `http://scikit-learn.org/`

[4] `https://radimrehurek.com/gensim/models/word2vec.html`

5.3 GloVe

Global Vectors for Word Representation (GloVe) uses stochastic gradient updates but operates on a "global" representation of word co-occurrence that is calculated once at the beginning of the algorithm (Pennington et al., 2014). Words and contexts are associated with bias parameters, b_w and b_c, where w is a word and c is a context, learned by minimizing the cost function:

$$\mathcal{L} = \sum_{w,c} f(x_{wc})\vec{w} \cdot \vec{c} + b_w + b_c - log(x_{wc}).$$

We use the GloVe implementation provided by Pennington et al. (2014)[5]. We use the default settings provided with GloVe except as described above.

Unlike SGNS, the algorithm does not perform model updates while examining the original documents. As a result, we expect GloVe to be sensitive to random initializations but not sensitive to the order of documents.

5.4 PPMI

The positive pointwise mutual information (PPMI) matrix, whose cells represent the PPMI of each pair of words and contexts, is factored using singular value decomposition (SVD) and results in low-dimensional embeddings that perform similarly to GloVe and SGNS (Levy and Goldberg, 2014).

$$PMI(w, c) = log\frac{P(w, c)}{P(w)P(c)};$$

$$PPMI(w, c) = max(PMI(w, c), 0).$$

To train our PPMI word embeddings, we use `hyperwords`,[6] an implementation provided as part of Levy et al. (2015).[7] We follow the authors' recommendations and set the context distributional smoothing (`cds`) parameter to 0.75, the eigenvalue matrix (`eig`) to 0.5, the subsampling threshold (`sub`) to 10^{-5}, and the context window (`win`) to 5.

[5] `http://nlp.stanford.edu/projects/glove/`

[6] `https://bitbucket.org/omerlevy/hyperwords/src`

[7] We altered the PPMI code to remove a fixed random seed in order to introduce variability given a fixed corpus; no other change was made.

Like GloVe and unlike SGNS, PPMI operates on a pre-computed representation of word co-occurrence, so we do not expect results to vary based on the order of documents. Unlike both GloVe and SGNS, PPMI uses a stable, non-stochastic SVD algorithm that should produce the same result given the same input, regardless of initialization. However, we expect variation due to PPMI's random subsampling of frequent tokens.

6 Methods

To establish statistical significance bounds for our observations, we train 50 LSA models, 50 SGNS models, 50 GloVe models, and 50 PPMI models for each of the three settings (FIXED, SHUFFLED, and BOOTSTRAP), for each document segmentation size, for each corpus.

For each corpus, we select a set of 20 relevant query words from high probability words from an LDA topic model (Blei et al., 2003) trained on that corpus with 200 topics. We calculate the cosine similarity of each query word to the other words in the vocabulary, creating a similarity ranking of all the words in the vocabulary. We calculate the mean and standard deviation of the cosine similarities for each pair of query word and vocabulary word across each set of 50 models.

From the lists of queries and cosine similarities, we select the 20 words most closely related to the set of query words and compare the mean and standard deviation of those pairs across settings. We calculate the Jaccard similarity between top-N lists to compare membership change in the lists of most closely related words, and we find average changes in rank within those lists. We examine these metrics across different algorithms and corpus parameters.

7 Results

We begin with a case study of the framing around the query term `marijuana`. One might hypothesize that the authors of various corpora (e.g. judges of the 4th Circuit, journalists at the NYT, and users on Reddit) have different perceptions of this drug and that their language might reflect those differences. Indeed, after qualitatively examining the lists of most similar terms (see Table 4), we might come to the conclusion that the allegedly conservative 4th Circuit judges view marijuana as similar to illegal drugs such as heroin and cocaine, while Reddit users view marijuana as closer to legal substances such as nicotine and alcohol.

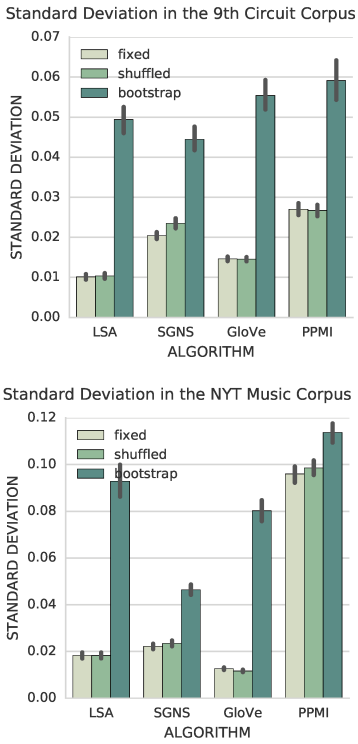

Figure 1: The mean standard deviations across settings and algorithms for the 10 closest words to the query words in the *9th Circuit* and *NYT Music* corpora using the whole documents. Larger variations indicate less stable embeddings.

However, we observe patterns that cause us to lower our confidence in such conclusions. Table 4 shows that the cosine similarities can vary significantly. We see that the top ranked words (chosen according to their mean cosine similarity across runs of the FIXED setting) can have widely different mean similarities and standard deviations depending on the algorithm and the three training settings, FIXED, SHUFFLED, and BOOTSTRAP.

As expected, each algorithm has a small variation in the FIXED setting. For example, we can see the effect of the random SVD solver for LSA and the effect of random subsampling for PPMI. We do not observe a consistent effect for document order in the SHUFFLED setting.

Most importantly, these figures reveal that the

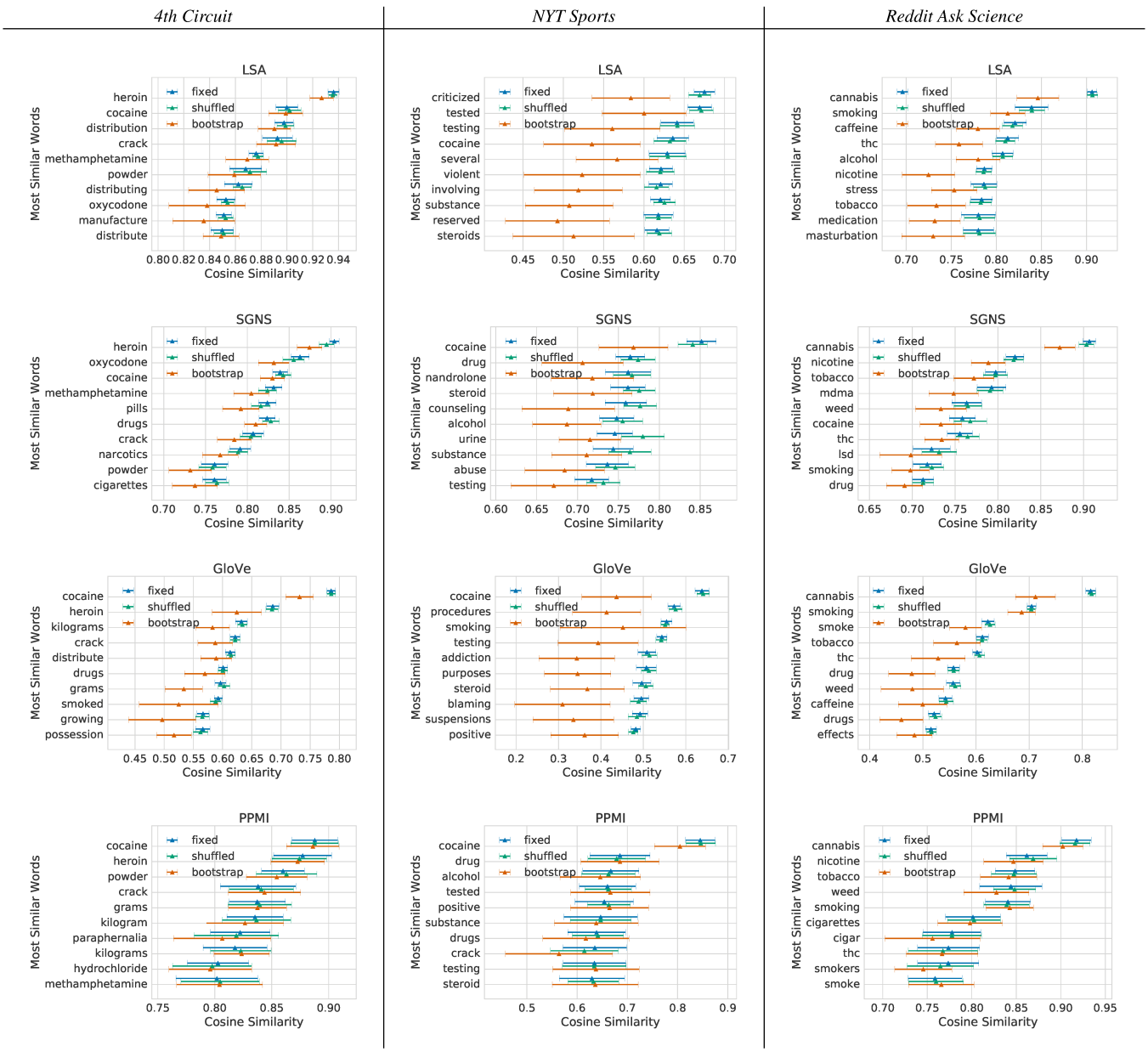

Table 4: The most similar words with their means and standard deviations for the cosine similarities between the query word `marijuana` and its 10 nearest neighbors (highest mean cosine similarity in the FIXED setting. Embeddings are learned from documents segmented by sentence.

BOOTSTRAP setting causes large increases in variation across all algorithms (with a weaker effect for PPMI) and corpora, with large standard deviations across word rankings. This indicates that the presence of specific documents in the corpus can significantly affect the cosine similarities between embedding vectors.

GloVe produced very similar embeddings in both the FIXED and SHUFFLED settings, with similar means and small standard deviations, which indicates that GloVe is not sensitive to document order. However, the BOOTSTRAP setting caused a reduction in the mean and widened the standard deviation, indicating that GloVe is sensitive to the presence of specific documents.

Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7
viability	fetus	trimester	surgery	trimester	pregnancies	abdomen
pregnancies	pregnancies	surgery	visit	surgery	occupation	tenure
abortion	gestation	visit	therapy	incarceration	viability	stepfather
abortions	kindergarten	tenure	pain	visit	abortion	wife
fetus	viability	workday	hospitalization	arrival	tenure	groin
gestation	headaches	abortions	neck	pain	visit	throat
surgery	pregnant	hernia	headaches	headaches	abortions	grandmother
expiration	abortion	summer	trimester	birthday	pregnant	daughter
sudden	pain	suicide	experiencing	neck	birthday	panic
fetal	bladder	abortion	medications	tenure	fetus	jaw

Table 5: The 10 closest words to the query term `pregnancy` are highly variable. None of the words shown appear in every run. Results are shown across runs of the BOOTSTRAP setting for the full corpus of the *9th Circuit*, the whole document size, and the SGNS model.

Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7
selection	selection	selection	selection	selection	selection	selection
genetics	process	human	darwinian	convergent	evolutionary	darwinian
convergent	darwinian	humans	theory	darwinian	humans	nature
process	humans	natural	genetics	evolutionary	species	evolutionary
darwinian	convergent	genetics	human	genetics	convergent	convergent
abiogenesis	evolutionary	species	evolutionary	theory	process	process
evolutionary	species	did	humans	natural	natural	natural
natural	human	convergent	natural	humans	did	species
nature	natural	process	convergent	process	human	humans
species	theory	evolutionary	creationism	human	darwinian	favor

Table 6: The order of the 10 closest words to the query term `evolution` are highly variable. Results are shown across runs of the BOOTSTRAP setting for the full corpus of *AskScience*, the whole document length, and the GloVe model.

These patterns of larger or smaller variations are generalized in Figure 1, which shows the mean standard deviation for different algorithms and settings. We calculated the standard deviation across the 50 runs for each query word in each corpus, and then we averaged over these standard deviations. The results show the average levels of variation for each algorithm and corpus. We observe that the FIXED and SHUFFLED settings for GloVe and LSA produce the least variable cosine similarities, while PPMI produces the most variable cosine similarities for all settings. The presence of specific documents has a significant effect on all four algorithms (lesser for PPMI), consistently increasing the standard deviations.

We turn to the question of how this variation in standard deviation affects the lists of most similar words. Are the top-N words simply re-ordered, or do the words present in the list substantially change? Table 5 shows an example of the top-N word lists for the query word `pregnancy` in the *9th Circuit* corpus. Observing Run 1, we might believe that judges of the 9th Circuit associate pregnancy most with questions of viability and abortion, while observing Run 5, we might believe that pregnancy is most associated with questions of prisons and family visits. Although the lists in this table are all produced from the same corpus and document size, the membership of the lists changes substantially between runs of the BOOTSTRAP setting.

As another example, Table 6 shows results for the query `evolution` for the GloVe model and the *AskScience* corpus. Although this query shows less variation between runs, we still find cause for concern. For example, Run 3 ranks the words `human` and `humans` highly, while Run 1 includes neither of those words in the top 10.

These changes in top-N rank are shown in Figure 2. For each query word for the *AskHistorians* corpus, we find the N most similar words using SGNS. We generate new top-N lists for each of the 50 models trained in the BOOTSTRAP setting, and we use Jaccard similarity to compare the 50 lists. We observe similar patterns to the changes in standard deviation

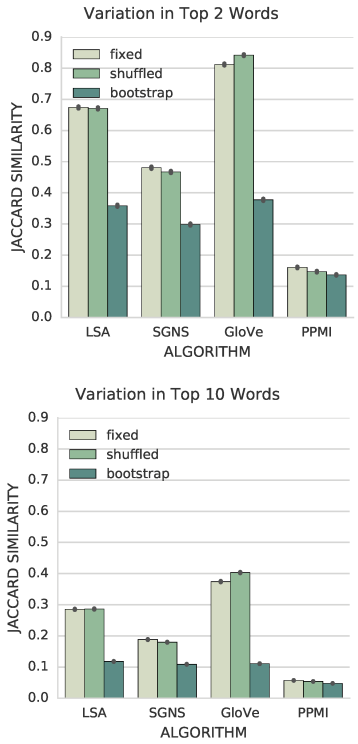

Figure 2: The mean Jaccard similarities across settings and algorithms for the top 2 and 10 closest words to the query words in the *AskHistorians* corpus. Larger Jaccard similarity indicates more consistency in top N membership. Results are shown for the sentence document length.

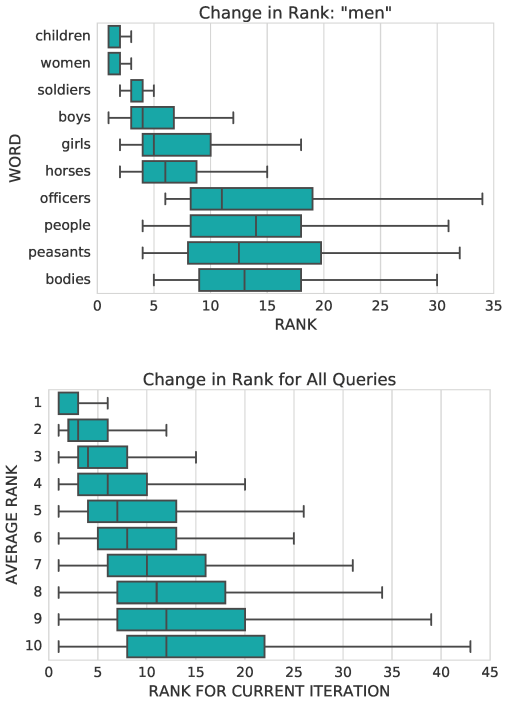

Figure 3: The change in rank across runs of the BOOTSTRAP setting for the top 10 words. We show results for both a single query, `men`, and an aggregate of all the queries, showing the change in rank of the words whose average ranking falls within the 10 nearest neighbors of those queries. Results are shown for SGNS on the *AskHistorians* corpus and the sentence document length.

in Figure 2; PPMI displays the lowest Jaccard similarity across settings, while the other algorithms have higher similarities in the FIXED and SHUFFLED settings but much lower similarities in the BOOTSTRAP setting. We display results for both $N = 2$ and $N = 10$, emphasizing that even very highly ranked words often drop out of the top-N list.

Even when words do not drop out of the top-N list, they often change in rank, as we observe in Figure 3. We show both a specific example for the query term `men` and an aggregate of all the terms whose average rank is within the top-10 across runs of the BOOTSTRAP setting. In order to highlight the average changes in rank, we do not show outliers in this figure, but we note that outliers (large falls and jumps in rank) are common. The variability across samples from the BOOTSTRAP setting indicates that the presence of specific documents can significantly affect the top-N rankings.

We also find that document segmentation size affects the cosine similarities. Figure 4 shows that documents segmented at a more fine-grained level produce embeddings with less variability across runs of the BOOTSTRAP setting. Documents segmented at the sentence level have standard deviations clustering closer to the median, while larger documents have standard deviations that are spread more widely. This effect is most significant for the *4th Circuit* and *9th Circuit* corpora, as these have much larger "documents" than the other corpora. We observe a similar effect for corpus size in Figure 5. The smaller corpus shows a larger spread in standard deviation than the larger corpus, indicating greater variability.

Finally, we find that the variance usually stabilizes at about 25 runs of the BOOTSTRAP setting. Figure 6 shows that variability initially increases with the number of models trained. We observe this pattern across corpora, algorithms, and settings.

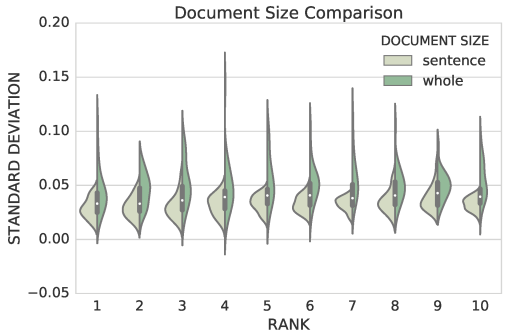

Figure 4: Standard deviation of the cosine similarities between all rank N words and their 10 nearest neighbors. Results are shown for different document sizes (sentence vs whole document) in the BOOTSTRAP setting for SGNS in the *4th Circuit* corpus.

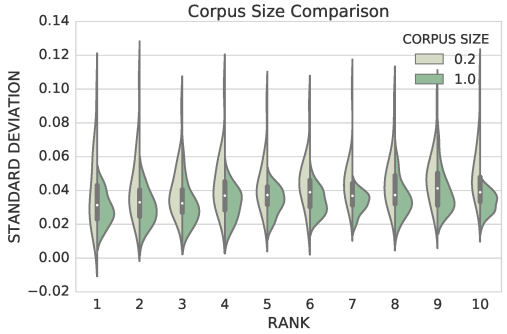

Figure 5: Standard deviation of the cosine similarities between all rank N words and their 10 nearest neighbors. Results are shown at different corpus sizes (20% vs 100% of documents) in the BOOTSTRAP setting for SGNS in the *4th Circuit* corpus, segmented by sentence.

8 Discussion

The most obvious result of our experiments is to emphasize that embeddings are not even a single objective view of a corpus, much less an objective view of language. The corpus is itself only a sample, and we have shown that the curation of this sample (its size, document length, and inclusion of specific documents) can cause significant variability in the embeddings. Happily, this variability can be quantified by averaging results over multiple bootstrap samples.

We can make several specific observations about algorithm sensitivities. In general, LSA, GloVe, SGNS, and PPMI are not sensitive to document order in the collections we evaluated. This is surprising, as we had expected SGNS to be sensitive to document order and anecdotally, we had observed cases where the embeddings were affected by groups of documents (e.g. in a different language) at the beginning of training. However, all four algorithms are sensitive to the presence of specific documents, though this effect is weaker for PPMI.

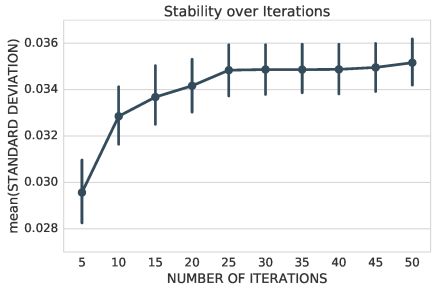

Figure 6: The mean of the standard deviation of the cosine similarities between each query term and its 20 nearest neighbors. Results are shown for different numbers of runs of the BOOTSTRAP setting on the *4th Circuit* corpus.

Although PPMI appears deterministic (due to its pre-computed word-context matrix), we find that this algorithm produced results under the FIXED ordering whose variability was closest to the BOOTSTRAP setting. We attribute this intrinsic variability to the use of token-level subsampling. This sampling method introduces variation into the source corpus that appears to be comparable to a bootstrap resampling method. Sampling in PPMI is inspired by a similar method in the word2vec implementation of SGNS (Levy et al., 2015). It is therefore surprising that SGNS shows noticeable differentiation between the BOOTSTRAP setting on the one hand and the FIXED and SHUFFLED settings on the other.

The use of embeddings as sources of evidence needs to be tempered with the understanding that fine-grained distinctions between cosine similarities are not reliable and that smaller corpora and longer documents are more susceptible to variation in the cosine similarities between embeddings. When studying the top-N most similar words to a query, it is important to account for variation in these lists, as both rank and membership can significantly change across runs. Therefore, we emphasize that with smaller corpora comes greater variability, and we recommend that practitioners use bootstrap sampling to generate an

ensemble of word embeddings for each sub-corpus and present both the mean and variability of any summary statistics such as ordered word similarities.

We leave for future work a full hyperparameter sweep for the three algorithms. While these hyperparameters can substantially impact performance, our goal with this work was not to achieve high performance but to examine how the algorithms respond to changes in the corpus. We make no claim that one algorithm is better than another.

9 Conclusion

We find that there are several sources of variability in cosine similarities between word embeddings vectors. The size of the corpus, the length of individual documents, and the presence or absence of specific documents can all affect the resulting embeddings. While differences in word association are measurable and are often significant, small differences in cosine similarity are not reliable, especially for small corpora. If the intention of a study is to learn about a specific corpus, we recommend that practitioners test the statistical confidence of similarities based on word embeddings by training on multiple bootstrap samples.

10 Acknowledgements

This work was supported by NSF #1526155, #1652536, and the Alfred P. Sloan Foundation. We would like to thank Alexandra Schofield, Laure Thompson, our Action Editor Ivan Titov, and our anonymous reviewers for their helpful comments.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning research*, 3(Jan):993–1022.

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 *NIPS*, pages 4349–4357.

Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer.

Andreas Broscheid. 2011. Comparing circuits: Are some U.S. Courts of Appeals more liberal or conservative than others? *Law & Society Review*, 45(1), March.

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. In *ACL*.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for Twitter named entity recognition. In *HLT-NAACL*, pages 735–745.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. *HLT-ACL*, pages 1606–1615.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42.

Yoav Goldberg. 2017. *Neural Network Methods for Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. September, 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *Technical Report No. 2009-05*. Applied & Computational Mathematics, California Institute of Technology.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *ACL*.

Johannes Hellrich and Udo Hahn. 2016. Bad company–neighborhoods in neural embedding spaces considered harmful. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2785–2796.

Ryan Heuser. 2016. Word vectors in the eighteenth-century. In *IPAM Workshop: Cultural Analytics*.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science,*.

Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2016. Freshman or fresher? Quantifying the geographic variation of language in online social media. In *ICWSM*, pages 615–618.

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(2):211.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185.

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

Rasmus E. Madsen, David Kauchak, and Charles Elkan. 2005. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 545–552. ACM.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. *HLT-NAACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.

Lawrence Phillips, Kyle Shaffer, Dustin Arendt, Nathan Hodas, and Svitlana Volkova. 2017. Intrinsic and extrinsic evaluation of spatiotemporal text representations in Twitter streams. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 201–210.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. LDC2008T19. Linguistic Data Consortium.

Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. 2016. Swivel: Improving Embeddings by Noticing What's Missing. *arXiv:1602.02215*.

Yingtao Tian, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2016. On the convergent properties of word embedding methods. *arXiv preprint arXiv:1605.03956*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the ACL*, pages 384–394. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the ACL*, pages 719–725. ACL.

Justine Zhang, William L. Hamilton, Cristian Danescu-Niculescu-Mizil, Dan Jurafsky, and Jure Leskovec. 2017. Community identity and user engagement in a multi-community landscape. *Proceedings of ICWSM*.