Linear Algebraic Structure of Word Senses, with Applications to Polysemy

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, Andrej Risteski
Computer Science Department, Princeton University
35 Olden St, Princeton, NJ 08540
{arora,yuanzhil,yingyul,tengyu,risteski}@cs.princeton.edu

Abstract

Word embeddings are ubiquitous in NLP and information retrieval, but it is unclear what they represent when the word is polysemous. Here it is shown that multiple word senses reside in linear superposition *within* the word embedding and simple sparse coding can recover vectors that approximately capture the senses. The success of our approach, which applies to several embedding methods, is mathematically explained using a variant of the *random walk on discourses* model (Arora et al., 2016). A novel aspect of our technique is that each extracted word sense is accompanied by one of about 2000 "discourse atoms" that gives a succinct description of which other words co-occur with that word sense. Discourse atoms can be of independent interest, and make the method potentially more useful. Empirical tests are used to verify and support the theory.

1 Introduction

Word embeddings are constructed using Firth's hypothesis that a word's sense is captured by the distribution of other words around it (Firth, 1957). Classical vector space models (see the survey by Turney and Pantel (2010)) use simple linear algebra on the matrix of word-word co-occurrence counts, whereas recent neural network and energy-based models such as word2vec use an objective that involves a nonconvex (thus, also nonlinear) function of the word co-occurrences (Bengio et al., 2003; Mikolov et al., 2013a; Mikolov et al., 2013b).

This nonlinearity makes it hard to discern how these modern embeddings capture the different senses of a polysemous word. The monolithic view of embeddings, with the internal information extracted only via inner product, is felt to fail in capturing word senses (Griffiths et al., 2007; Reisinger and Mooney, 2010; Iacobacci et al., 2015). Researchers have instead sought to capture polysemy using more complicated representations, e.g., by inducing separate embeddings for each sense (Murphy et al., 2012; Huang et al., 2012). These embedding-per-sense representations grow naturally out of classic Word Sense Induction or WSI (Yarowsky, 1995; Schutze, 1998; Reisinger and Mooney, 2010; Di Marco and Navigli, 2013) techniques that perform clustering on neighboring words.

The current paper goes beyond this monolithic view, by describing how multiple senses of a word actually reside in linear superposition *within* the standard word embeddings (e.g., word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014)). By this we mean the following: consider a polysemous word, say *tie*, which can refer to an article of clothing, or a drawn match, or a physical act. Let's take the usual viewpoint that *tie* is a single token that represents monosemous words *tie1, tie2, ...* The theory and experiments in this paper strongly suggest that word embeddings computed using modern techniques such as GloVe and word2vec satisfy:

$$v_{tie} \approx \alpha_1 \, v_{tie1} + \alpha_2 \, v_{tie2} + \alpha_3 \, v_{tie3} + \cdots \quad (1)$$

where coefficients α_i's are nonnegative and v_{tie1}, v_{tie2}, etc., are the hypothetical embeddings of

Transactions of the Association for Computational Linguistics, vol. 6, pp. 483–495, 2018. Action Editor: Hinrich Schütze.
Submission batch: 11/2017; Revision batch: 3/2018; Published 7/2018.

the different senses—those that *would* have been induced in the thought experiment where all occurrences of the different senses were hand-labeled in the corpus. This *Linearity Assertion*, whereby linear structure appears out of a highly nonlinear embedding technique, is explained theoretically in Section 2, and then empirically tested in a couple of ways in Section 4.

Section 3 uses the linearity assertion to show how to do WSI via sparse coding, which can be seen as a linear algebraic analog of the classic clustering-based approaches, albeit with overlapping clusters. On standard testbeds it is competitive with earlier embedding-for-each-sense approaches (Section 6). A novelty of our WSI method is that it automatically links different senses of different words via our *atoms of discourse* (Section 3). This can be seen as an answer to the suggestion in (Reisinger and Mooney, 2010) to enhance one-embedding-per-sense methods so that they can automatically link together senses for different words, e.g., recognize that the "article of clothing" sense of *tie* is connected to *shoe, jacket,* etc.

This paper is inspired by the solution of word analogies via linear algebraic methods (Mikolov et al., 2013b), and use of sparse coding on word embeddings to get useful representations for many NLP tasks (Faruqui et al., 2015). Our theory builds conceptually upon the *random walk on discourses* model of Arora et al. (2016), although we make a small but important change to explain empirical findings regarding polysemy. Our WSI procedure applies (with minor variation in performance) to canonical embeddings such as word2vec and GloVe as well as the older vector space methods such as PMI (Church and Hanks, 1990). This is not surprising since these embeddings are known to be interrelated (Levy and Goldberg, 2014; Arora et al., 2016).

2 Justification for Linearity Assertion

Since word embeddings are solutions to nonconvex optimization problems, at first sight it appears hopeless to reason about their finer structure. But it becomes possible to do so using a generative model for language (Arora et al., 2016) — a dynamic versions by the log-linear topic model of (Mnih and Hinton, 2007)—which we now recall. It posits that at every point in the corpus there is a micro-topic ("what is being talked about") called *discourse* that is drawn from the continuum of unit vectors in $\Re^d$. The parameters of the model include a vector $v_w \in \Re^d$ for each word w. Each discourse c defines a distribution over words $\Pr[w \mid c] \propto \exp(c \cdot v_w)$. The model assumes that the corpus is generated by the slow geometric random walk of c over the unit sphere in $\Re^d$: when the walk is at c, a few words are emitted by i.i.d. samples from the distribution (2), which, due to its log-linear form, strongly favors words close to c in cosine similarity. Estimates for learning parameters v_w using MLE and moment methods correspond to standard embedding methods such as GloVe and word2vec (see the original paper).

To study how word embeddings capture word senses, we'll need to understand the relationship between a word's embedding and those of words it co-occurs with. In the next subsection, we propose a slight modification to the above model and shows how to infer the embedding of a word from the embeddings of other words that co-occur with it. This immediately leads to the Linearity Assertion, as shown in Section 2.2.

2.1 Gaussian Walk Model

As alluded to before, we modify the random walk model of (Arora et al., 2016) to the *Gaussian random walk model*. Again, the parameters of the model include a vector $v_w \in \Re^d$ for each word w. The model assumes the corpus is generated as follows. First, a discourse vector c is drawn from a Gaussian with mean 0 and covariance Σ. Then, a window of n words $w_1, w_2, \dots, w_n$ are generated from c by:

$$\Pr[w_1, w_2, \dots, w_n| \, c] = \prod_{i=1}^{n} \Pr[w_i| \, c], \tag{2}$$

$$\Pr[w_i \mid c] = \exp(c \cdot v_{w_i})/Z_c, \tag{3}$$

where $Z_c = \sum_w \exp(\langle v_w, c\rangle)$ is the partition function. We also assume the partition function concentrates in the sense that $Z_c \approx Z \exp(\|c\|^2)$ for some constant Z. This is a direct extension of (Arora et al., 2016, Lemma 2.1) to discourse vectors with norm other than 1, and causes the additional term $\exp(\|c\|^2)$.[1]

[1] The formal proof of (Arora et al., 2016) still applies in this setting. The simplest way to informally justify this assumption

Theorem 1. *Assume the above generative model, and let s denote the random variable of a window of n words. Then, there is a linear transformation A such that $v_w \approx A\,\mathbb{E}\left[\frac{1}{n}\sum_{w_i\in s} v_{w_i} \mid w \in s\right]$.*

Proof. Let c_s be the discourse vector for the whole window s. By the law of total expectation, we have

$$\begin{aligned}&\mathbb{E}\left[c_s \mid w \in s\right]\\ =&\mathbb{E}\left[\mathbb{E}[c_s \mid s = w_1 \dots w_{j-1} w w_{j+1} \dots w_n] \mid w \in s\right].\end{aligned} \tag{4}$$

We evaluate the two sides of the equation.

First, by Bayes' rule and the assumptions on the distribution of c and the partition function, we have:

$$\begin{aligned}p(c|w) &\propto p(w|c)p(c)\\ &\propto \frac{1}{Z_c}\exp(\langle v_w, c\rangle)\cdot\exp\left(-\frac{1}{2}c^\top\Sigma^{-1}c\right)\\ &\approx \frac{1}{Z}\exp\left(\langle v_w, c\rangle - c^\top\left(\frac{1}{2}\Sigma^{-1}+I\right)c\right).\end{aligned}$$

So $c \mid w$ is a Gaussian distribution with mean

$$\mathbb{E}\left[c \mid w\right] \approx (\Sigma^{-1}+2I)^{-1}v_w. \tag{5}$$

Next, we compute $\mathbb{E}[c|w_1,\dots,w_n]$. Again using Bayes' rule and the assumptions on the distribution of c and the partition function,

$$\begin{aligned}&p(c|w_1,\dots,w_n)\\ &\propto p(w_1,\dots,w_n|c)p(c)\\ &\propto p(c)\prod_{i=1}^{n}p(w_i|c)\\ &\approx \frac{1}{Z^n}\exp\left(\sum_{i=1}^{n}v_{w_i}^\top c - c^\top\left(\frac{1}{2}\Sigma^{-1}+nI\right)c\right).\end{aligned}$$

So $c|w_1\dots w_n$ is a Gaussian distribution with mean

$$\mathbb{E}[c|w_1,\dots,w_n] \approx \left(\Sigma^{-1}+2nI\right)^{-1}\sum_{i=1}^{n}v_{w_i}. \tag{6}$$

Now plugging in equation (5) and (6) into equation (4), we conclude that

$$(\Sigma^{-1}+2I)^{-1}v_w \approx (\Sigma^{-1}+2nI)^{-1}\mathbb{E}\left[\sum_{i=1}^{n}v_{w_i} \mid w \in s\right].$$

Re-arranging the equation completes the proof with $A = n(\Sigma^{-1}+2I)(\Sigma^{-1}+2nI)^{-1}$. □

Note: Interpretation. Theorem 1 shows that there exists a linear relationship between the vector of a word and the vectors of the words in its contexts. Consider the following thought experiment. First, choose a word w. Then, for each window s containing w, take the average of the vectors of the words in s and denote it as v_s. Now, take the average of v_s for all the windows s containing w, and denote the average as u. Theorem 1 says that u can be mapped to the word vector v_w by a linear transformation that does not depend on w. This linear structure may also have connections to some other phenomena related to linearity, e.g., Gittens et al. (2017) and Tian et al. (2017). Exploring such connections is left for future work.

The linear transformation is closely related to Σ, which describes the distribution of the discourses. If we choose a coordinate system such that Σ is a diagonal matrix with diagonal entries λ_i, then A will also be a diagonal matrix with diagonal entries $(n+2n\lambda_i)/(1+2n\lambda_i)$. This is smoothing the spectrum and essentially shrinks the directions corresponding to large λ_i relatively to the other directions. Such directions are for common discourses and thus common words. Empirically, we indeed observe that A shrinks the directions of common words. For example, its last right singular vector has, as nearest neighbors, the vectors for words like "with", "as", and "the." Note that empirically, A is not a diagonal matrix since the word vectors are not in the coordinate system mentioned.

Note: Implications for GloVe and word2vec. Repeating the calculation in Arora et al. (2016) for our new generative model, we can show that the solutions to GloVe and word2vec training objectives solve for the following vectors: $\hat{v}_w = \left(\Sigma^{-1}+4I\right)^{-1/2}v_w$. Since these other embeddings are the same as v_w's up to linear transformation, Theorem 1 (and the Linearity Assertion) still holds for them. Empirically, we find that $\left(\Sigma^{-1}+4I\right)^{-1/2}$ is close to a scaled identity matrix (since $\|\Sigma^{-1}\|_2$ is small), so $\hat{v}_w$'s can be used as a surrogate of v_w's.

Experimental note: Using better sentence embeddings, SIF embeddings. Theorem 1 implicitly uses the average of the neighboring word vectors as

is to assume v_w are random vectors, and then Z_c can be shown to concentrate around $\exp(\|c\|^2)$. Such a condition enforces the word vectors to be isotropic to some extent, and makes the covariance of the discourse identifiable.

an estimate (MLE) for the discourse vector. This estimate is of course also a simple *sentence embedding*, very popular in empirical NLP work and also reminiscent of word2vec's training objective. In practice, this naive sentence embedding can be improved by taking a weighted combination (often tf-idf) of adjacent words. The paper (Arora et al., 2017) uses a simple twist to the generative model in (Arora et al., 2016) to provide a better estimate of the discourse c called SIF embedding, which is better for downstream tasks and surprisingly competitive with sophisticated LSTM-based sentence embeddings. It is a *weighted* average of word embeddings in the window, with smaller weights for more frequent words (reminiscent of tf-idf). This weighted average is the MLE estimate of c if above generative model is changed to:

$$p(w|c) = \alpha p(w) + (1-\alpha)\frac{\exp(v_w \cdot c)}{Z_c},$$

where $p(w)$ is the overall probability of word w in the corpus and $\alpha > 0$ is a constant (hyperparameter).

The theory in the current paper works with SIF embeddings as an estimate of the discourse c; in other words, in Theorem 1 we replace the average word vector with the SIF vector of that window. Empirically we find that it leads to similar results in testing our theory (Section 4) and better results in downstream WSI applications (Section 6). Therefore, SIF embeddings are adopted in our experiments.

2.2 Proof of Linearity Assertion

Now we use Theorem 1 to show how the Linearity Assertion follows. Recall the thought experiment considered there. Suppose word w has two distinct senses s_1 and s_2. Compute a word embedding v_w for w. Then hand-replace each occurrence of a sense of w by one of the new tokens s_1, s_2 depending upon which one is being used. Next, train separate embeddings for s_1, s_2 while keeping the other embeddings fixed. (NB: the classic clustering-based sense induction (Schutze, 1998; Reisinger and Mooney, 2010) can be seen as an approximation to this thought experiment.)

Theorem 2 (Main). *Assuming the model of Section 2.1, embeddings in the thought experiment above will satisfy* $\|v_w - \bar{v}_w\|_2 \to 0$ *as the corpus length tends to infinity, where* $\bar{v}_w \approx \alpha v_{s_1} + \beta v_{s_2}$ *for*

$$\alpha = \frac{f_1}{f_1 + f_2}, \quad \beta = \frac{f_2}{f_1 + f_2},$$

where f_1 *and* f_2 *are the numbers of occurrences of* s_1, s_2 *in the corpus, respectively.*

Proof. Suppose we pick a random sample of N windows containing w in the corpus. For each window, compute the average of the word vectors and then apply the linear transformation in Theorem 1. The transformed vectors are i.i.d. estimates for v_w, but with high probability about $f_1/(f_1+f_2)$ fraction of the occurrences used sense s_1 and $f_2/(f_1+f_2)$ used sense s_2, and the corresponding estimates for those two subpopulations converge to v_{s_1} and v_{s_2} respectively. Thus by construction, the estimate for v_w is a linear combination of those for v_{s_1} and v_{s_2}. □

Note. Theorem 1 (and hence the Linearity Assertion) holds already for the original model in Arora et al. (2016) but with $A = I$, where I is the identity transformation. In practice, we find inducing the word vector requires a non-identity A, which is the reason for the modified model of Section 2.1. This also helps to address a nagging issue hiding in older clustering-based approaches such as Reisinger and Mooney (2010) and Huang et al. (2012), which identified senses of a polysemous word by clustering the sentences that contain it. One imagines a good representation of the sense of an individual cluster is simply the cluster center. This turns out to be false — the closest words to the cluster center sometimes are not meaningful for the sense that is being captured; see Table 1. Indeed, the authors of Reisinger and Mooney (2010) seem aware of this because they mention "We do not assume that clusters correspond to traditional word senses. Rather, we only rely on clusters to capture meaningful variation in word usage." We find that applying A to cluster centers makes them meaningful again. See also Table 1.

3 Towards WSI: Atoms of Discourse

Now we consider how to do WSI using only word embeddings and the Linearity Assertion. Our approach is fully unsupervised, and tries to induce senses for all words in one go, together with a vector representation for each sense.

center 1	before	and provide providing a
	after	providing provide opportunities provision
center 2	before	and a to the
	after	access accessible allowing provide

Table 1: Four nearest words for some cluster centers that were computed for the word "access" by applying 5-means on the estimated discourse vectors (see Section 2.1) of 1000 random windows from Wikipedia containing "access". After applying the linear transformation of Theorem 1 to the center, the nearest words become meaningful.

Given embeddings for all words, it seems unclear at first sight how to pin down the senses of *tie* using only (1) since v_{tie} can be expressed in infinitely many ways as such a combination, and this is true even if α_i's were known (and they aren't). To pin down the senses we will need to interrelate the senses of different words, for example, relate the "article of clothing" sense *tie1* with *shoe, jacket,* etc. To do so we rely on the generative model of Section 2.1 according to which unit vector in the embedding space corresponds to a micro-topic or discourse. Empirically, discourses c and c' tend to look similar to humans (in terms of nearby words) if their inner product is larger than 0.85, and quite different if the inner product is smaller than 0.5. So in the discussion below, a discourse should really be thought of as a small region rather than a point.

One imagines that the corpus has a "clothing" discourse that has a high probability of outputting the *tie1* sense, and also of outputting related words such as *shoe, jacket,* etc. By (2) the probability of being output by a discourse is determined by the inner product, so one expects that the vector for "clothing" discourse has a high inner product with all of *shoe, jacket, tie1,* etc., and thus can stand as surrogate for v_{tie1} in (1)! Thus it may be sufficient to consider the following global optimization:

Given word vectors $\{v_w\}$ in $\Re^d$ and two integers k, m with $k < m$, find a set of unit vectors $A_1, A_2, \ldots, A_m$ such that

$$v_w = \sum_{j=1}^{m} \alpha_{w,j} A_j + \eta_w \tag{7}$$

where at most k of the coefficients $\alpha_{w,1}, \ldots, \alpha_{w,m}$ are nonzero, and η_w's are error vectors.

Here k is the sparsity parameter, and m is the number of atoms, and the optimization minimizes the norms of η_w's (the ℓ_2-reconstruction error):

$$\sum_w \left\| v_w - \sum_{j=1}^{m} \alpha_{w,j} A_j \right\|_2^2. \tag{8}$$

Both A_j's and $\alpha_{w,j}$'s are unknowns, and the optimization is nonconvex. This is just *sparse coding*, useful in neuroscience (Olshausen and Field, 1997) and also in image processing, computer vision, etc.

This optimization is a surrogate for the desired expansion of v_{tie} as in (1), because one can hope that among $A_1, \ldots, A_m$ there will be directions corresponding to *clothing*, *sports matches,* etc., that will have high inner products with *tie1*, *tie2,* etc., respectively. Furthermore, restricting m to be much smaller than the number of words ensures that the typical A_i needs to be reused to express multiple words.

We refer to A_i's, discovered by this procedure, as *atoms of discourse*, since experimentation suggests that the actual discourse in a typical place in text (namely, vector c in (2)) is a linear combination of a small number, around 3-4, of such atoms. Implications of this for text analysis are left for future work.

Relationship to Clustering. Sparse coding is solved using alternating minimization to find the A_i's that minimize (8). This objective function reveals sparse coding to be a linear algebraic analogue of *overlapping* clustering, whereby the A_i's act as *cluster centers* and each v_w is assigned in a soft way to at most k of them (using the coefficients $\alpha_{w,j}$, of which at most k are nonzero). In fact this clustering viewpoint is also the basis of the alternating minimization algorithm. In the special case when $k = 1$, each v_w has to be assigned to a single cluster, which is the familiar geometric clustering with squared ℓ_2 distance.

Similar overlapping clustering in a traditional graph-theoretic setup —clustering while simultaneously cross-relating the senses of different words—seems more difficult but worth exploring.

4 Experimental Tests of Theory

4.1 Test of Gaussian Walk Model: Induced Embeddings

Now we test the prediction of the Gaussian walk model suggesting a linear method to induce embed-

#paragraphs	250k	500k	750k	1 million
cos similarity	0.94	0.95	0.96	0.96

Table 2: Fitting the GloVe word vectors with average discourse vectors using a linear transformation. The first row is the number of paragraphs used to compute the discourse vectors, and the second row is the average cosine similarities between the fitted vectors and the GloVe vectors.

dings from the context of a word. Start with the GloVe embeddings; let v_w denote the embedding for w. Randomly sample many paragraphs from Wikipedia, and for each word w' and each occurrence of w' compute the SIF embedding of text in the window of 20 words centered around w'. Average the SIF embeddings for all occurrences of w' to obtain vector $u_{w'}$. The Gaussian walk model says that there is a linear transformation that maps $u_{w'}$ to $v_{w'}$, so solve the regression:

$$\text{argmin}_A \sum_w \|Au_w - v_w\|_2^2. \tag{9}$$

We call the vectors Au_w the *induced embeddings*. We can test this method of inducing embeddings by holding out 1/3 words randomly, doing the regression (9) on the rest, and computing the cosine similarities between Au_w and v_w on the heldout set of words.

Table 2 shows that the average cosine similarity between the induced embeddings and the GloVe vectors is large. By contrast the average similarity between the average discourse vectors and the GloVe vectors is much smaller (about 0.58), illustrating the need for the linear transformation. Similar results are observed for the word2vec and SN vectors (Arora et al., 2016).

4.2 Test of Linearity Assertion

We do two empirical tests of the Linearity Assertion (Theorem 2).

Test 1. The first test involves the classic artificial polysemous words (also called pseudowords). First, pre-train a set W_1 of word vectors on Wikipedia with existing embedding methods. Then, randomly pick m pairs of non-repeated words, and for each pair, replace each occurrence of either of the two words with a pseudoword. Third, train a set W_2 of vectors on the new corpus, while holding fixed the vectors of words that were not involved in the pseudowords. Construction has ensured that each pseudoword has two distinct "senses", and we also have in W_1 the "ground truth" vectors for those senses.[2] Theorem 2 implies that the embedding of a pseudoword is a linear combination of the sense vectors, so we can compare this predicted embedding to the one learned in W_2.[3]

m pairs		10	10^3	$3 \cdot 10^4$
relative error	SN	0.32	0.63	0.67
	GloVe	0.29	0.32	0.51
cos similarity	SN	0.90	0.72	0.75
	GloVe	0.91	0.91	0.77

Table 3: The average relative errors and cosine similarities between the vectors of pseudowords and those predicted by Theorem 2. m pairs of words are randomly selected and for each pair, all occurrences of the two words in the corpus is replaced by a pseudoword. Then train the vectors for the pseudowords on the new corpus.

Suppose the trained vector for a pseudoword w is u_w and the predicted vector is v_w, then the comparison criterion is the average relative error $\frac{1}{|S|}\sum_{w\in S}\frac{\|u_w - v_w\|_2^2}{\|v_w\|_2^2}$ where S is the set of all the pseudowords. We also report the average cosine similarity between v_w's and u_w's.

Table 3 shows the results for the GloVe and SN (Arora et al., 2016) vectors, averaged over 5 runs. When m is small, the error is small and the cosine similarity is as large as 0.9. Even if $m = 3 \cdot 10^4$

[2] Note that this discussion assumes that the set of pseudowords is small, so that a typical neighborhood of a pseudoword does not consist of other pseudowords. Otherwise the ground truth vectors in W_1 become a bad approximation to the sense vectors.

[3] Here W_2 is trained while fixing the vectors of words not involved in pseudowords to be their pre-trained vectors in W_1. We can also train all the vectors in W_2 from random initialization. Such W_2 will not be aligned with W_1. Then we can learn a linear transformation from W_2 to W_1 using the vectors for the words not involved in pseudowords, apply it on the vectors for the pseudowords, and compare the transformed vectors to the predicted ones. This is tested on word2vec, resulting in relative errors between 20% and 32%, and cosine similarities between 0.86 and 0.92. These results again support our analysis.

vector type	GloVe	skip-gram	SN
cosine	0.72	0.73	0.76

Table 4: The average cosine of the angles between the vectors of words and the span of vector representations of its senses. The words tested are those in the WSI task of SemEval 2010.

(i.e., about 90% of the words in the vocabulary are replaced by pseudowords), the cosine similarity remains above 0.7, which is significant in the 300 dimensional space. This provides positive support for our analysis.

Test 2. The second test is a proxy for what would be a complete (but laborious) test of the Linearity Assertion: replicating the thought experiment while hand-labeling sense usage for many words in a corpus. The simpler proxy is as follows. For each word w, WordNet (Fellbaum, 1998) lists its various senses by providing definition and example sentences for each sense. This is enough text (roughly a paragraph's worth) for our theory to allow us to represent it by a vector —specifically, apply the SIF sentence embedding followed by the linear transformation learned as in Section 4.1. The text embedding for sense s should approximate the ground truth vector v_s for it. Then the Linearity Assertion predicts that embedding v_w lies close to the subspace spanned by the sense vectors. (Note that this is a nontrivial event: in 300 dimensions a random vector will be quite far from the subspace spanned by some 3 other random vectors.) Table 4 checks this prediction using the polysemous words appearing in the WSI task of SemEval 2010. We tested three standard word embedding methods: GloVe, the skip-gram variant of word2vec, and SN (Arora et al., 2016). The results show that the word vectors are quite close to the subspace spanned by the senses.

5 Experiments with Atoms of Discourse

The experiments use 300-dimensional embeddings created using the SN objective in (Arora et al., 2016) and a Wikipedia corpus of 3 billion tokens (Wikimedia, 2012), and the sparse coding is solved by standard k-SVD algorithm (Damnjanovic et al., 2010). Experimentation showed that the best sparsity parameter k (i.e., the maximum number of allowed senses per word) is 5, and the number of atoms m is about 2000. For the number of senses k, we tried plausible alternatives (based upon suggestions of many colleagues) that allow k to vary for different words, for example to let k be correlated with the word frequency. But a fixed choice of $k = 5$ seems to produce just as good results. To understand why, realize that this method retains no information about the corpus except for the low dimensional word embeddings. Since the sparse coding tends to express a word using fairly different atoms, examining (7) shows that $\sum_j \alpha_{w,j}^2$ is bounded by approximately $\|v_w\|_2^2$. So if too many $\alpha_{w,j}$'s are allowed to be nonzero, then some must necessarily have small coefficients, which makes the corresponding components indistinguishable from noise. In other words, raising k often picks not only atoms corresponding to additional senses, but also many that don't.

The best number of atoms m was found to be around 2000. This was estimated by re-running the sparse coding algorithm multiple times with different random initializations, whereupon substantial overlap was found between the two bases: a large fraction of vectors in one basis were found to have a very close vector in the other. Thus combining the bases while merging duplicates yielded a basis of about the same size. Around 100 atoms are used by a large number of words or have no close-by words. They appear semantically meaningless and are excluded by checking for this condition.[4]

The content of each atom can be discerned by looking at the nearby words in cosine similarity. Some examples are shown in Table 5. Each word is represented using at most five atoms, which usually capture distinct senses (with some noise/mistakes). The senses recovered for *tie* and *spring* are shown in Table 6. Similar results can be obtained by using other word embeddings like word2vec and GloVe.

We also observe sparse coding procedures assign nonnegative values to most coefficients $\alpha_{w,j}$'s even if they are left unrestricted. Probably this is because the appearances of a word are best explained by what discourse *is* being used to generate it, rather than what discourses are *not* being used.

[4] We think semantically meaningless atoms —i.e., unexplained inner products—exist because a simple language model such as ours cannot explain all observed co-occurrences due to grammar, stopwords, etc. It ends up needing smoothing terms.

Atom 1978	825	231	616	1638	149	330
drowning	instagram	stakes	membrane	slapping	orchestra	conferences
suicides	twitter	thoroughbred	mitochondria	pulling	philharmonic	meetings
overdose	facebook	guineas	cytosol	plucking	philharmonia	seminars
murder	tumblr	preakness	cytoplasm	squeezing	conductor	workshops
poisoning	vimeo	filly	membranes	twisting	symphony	exhibitions
commits	linkedin	fillies	organelles	bowing	orchestras	organizes
stabbing	reddit	epsom	endoplasmic	slamming	toscanini	concerts
strangulation	myspace	racecourse	proteins	tossing	concertgebouw	lectures
gunshot	tweets	sired	vesicles	grabbing	solti	presentations

Table 5: Some discourse atoms and their nearest 9 words. By Equation (2), words most likely to appear in a discourse are those nearest to it.

tie					spring				
trousers	season	scoreline	wires	operatic	beginning	dampers	flower	creek	humid
blouse	teams	goalless	cables	soprano	until	brakes	flowers	brook	winters
waistcoat	winning	equaliser	wiring	mezzo	months	suspension	flowering	river	summers
skirt	league	clinching	electrical	contralto	earlier	absorbers	fragrant	fork	ppen
sleeved	finished	scoreless	wire	baritone	year	wheels	lilies	piney	warm
pants	championship	replay	cable	coloratura	last	damper	flowered	elk	temperatures

Table 6: Five discourse atoms linked to the words *tie* and *spring*. Each atom is represented by its nearest 6 words. The algorithm often makes a mistake in the last atom (or two), as happened here.

Relationship to Topic Models. Atoms of discourse may be reminiscent of results from other automated methods for obtaining a thematic understanding of text, such as topic modeling, described in the survey by Blei (2012). This is not surprising since the model (2) used to compute the embeddings is related to a log-linear topic model by Mnih and Hinton (2007). However, the discourses here are computed via sparse coding on word embeddings, which can be seen as a linear algebraic alternative, resulting in fairly fine-grained topics. Atoms are also reminiscent of coherent "word clusters" detected in the past using Brown clustering, or even sparse coding (Murphy et al., 2012). The novelty in this paper is a clear interpretation of the sparse coding results as atoms of discourse, as well as its use to capture different word senses.

6 Testing WSI in Applications

While the main result of the paper is to reveal the linear algebraic structure of word senses within existing embeddings, it is desirable to verify that this view can yield results competitive with earlier sense embedding approaches. We report some tests below. We find that common word embeddings perform similarly with our method; for concreteness we use induced embeddings described in Section 4.1. They are evaluated in three tasks: word sense induction task in SemEval 2010 (Manandhar et al., 2010), word similarity in context (Huang et al., 2012), and a new task we called police lineup test. The results are compared to those of existing embedding based approaches reported in related work (Huang et al., 2012; Neelakantan et al., 2014; Mu et al., 2017).

6.1 Word Sense Induction

In the WSI task in SemEval 2010, the algorithm is given a polysemous word and about 40 pieces of texts, each using it according to a single sense. The algorithm has to cluster the pieces of text so that those with the same sense are in the same cluster. The evaluation criteria are F-score (Artiles et al., 2009) and V-Measure (Rosenberg and Hirschberg, 2007). The F-score tends to be higher with a smaller number of clusters and the V-Measure tends to be higher with a larger number of clusters, and fair evaluation requires reporting both.

Given a word and its example texts, our algorithm uses a Bayesian analysis dictated by our theory to compute a vector u_c for each piece of text c and and then applies k-means on these vectors, with the small twist that sense vectors are assigned to nearest centers based on inner products rather than Euclidean distances. Table 7 shows the results.

Computing vector u_c. For word w we start by computing its expansion in terms of atoms of discourse (see (8) in Section 3). In an ideal world the nonzero coefficients would exactly capture its senses, and each text containing w would match to one of these nonzero coefficients. In the real world such deterministic success is elusive and one must reason using Bayes' rule.

For each atom a, word w and text c there is a joint distribution $p(w, a, c)$ describing the event that atom a is the sense being used when word w was used in text c. We are interested in the posterior distribution:

$$p(a|c, w) \propto p(a|w)p(a|c)/p(a). \tag{10}$$

We approximate $p(a|w)$ using Theorem 2, which suggests that the coefficients in the expansion of v_w with respect to atoms of discourse scale according to probabilities of usage. (This assertion involves ignoring the low-order terms involving the logarithm in the theorem statement.) Also, by the random walk model, $p(a|c)$ can be approximated by $\exp(\langle v_a, v_c \rangle)$ where v_c is the SIF embedding of the context. Finally, since $p(a) = \mathbf{E}_c[p(a|c)]$, it can be empirically estimated by randomly sampling c.

The posterior $p(a|c, w)$ can be seen as a soft decoding of text c to atom a. If texts c_1, c_2 both contain w, and they were hard decoded to atoms a_1, a_2 respectively then their similarity would be $\langle v_{a_1}, v_{a_2} \rangle$. With our soft decoding, the similarity can be defined by taking the expectation over the full posterior:

$$\begin{aligned} &\text{similarity}(c_1, c_2) \\ &= \mathbf{E}_{a_i \sim p(a|c_i, w), i \in \{1,2\}} \langle v_{a_1}, v_{a_2} \rangle, \qquad (11) \\ &= \left\langle \sum_{a_1} p(a_1|c_1, w) v_{a_1}, \sum_{a_2} p(a_2|c_2, w) v_{a_2} \right\rangle. \end{aligned}$$

At a high level this is analogous to the Bayesian polysemy model of Reisinger and Mooney (2010) and Brody and Lapata (2009), except that they introduced separate embeddings for each sense cluster, while here we are working with structure already existing inside word embeddings.

Method	V-Measure	F-Score
(Huang et al., 2012)	10.60	38.05
(Neelakantan et al., 2014)	9.00	47.26
(Mu et al., 2017), $k = 2$	7.30	**57.14**
(Mu et al., 2017), $k = 5$	**14.50**	44.07
ours, $k = 2$	6.1	**58.55**
ours, $k = 3$	7.4	55.75
ours, $k = 4$	9.9	51.85
ours, $k = 5$	**11.5**	46.38

Table 7: Performance of different vectors in the WSI task of SemEval 2010. The parameter k is the number of clusters used in the methods. Rows are divided into two blocks, the first of which shows the results of the competitors, and the second shows those of our algorithm. Best results in each block are in boldface.

The last equation suggests defining the vector u_c for the text c as

$$u_c = \sum_a p(a|c, w) v_a, \tag{12}$$

which allows the similarity between two text pieces to be expressed via the inner product of their vectors.

Results. The results are reported in Table 7. Our approach outperforms the results by Huang et al. (2012) and Neelakantan et al. (2014). When compared to Mu et al. (2017), for the case with 2 centers, we achieved better V-measure but lower F-score, while for 5 centers, we achieved lower V-measure but better F-score.

6.2 Word Similarity in Context

The dataset consists of around 2000 pairs of words, along with the contexts the words occur in and the ground-truth similarity scores. The evaluation criterion is the correlation between the ground-truth scores and the predicted ones. Our method computes the estimated sense vectors and then the similarity as in Section 6.1. We compare to the baselines that simply use the cosine similarity of the GloVe/skip-gram vectors, and also to the results of several existing sense embedding methods.

Results. Table 8 shows that our result is better than those of the baselines and Mu et al. (2017), but slightly worse than that of Huang et al. (2012).

Method	Spearman coefficient
GloVe	0.573
skip-gram	0.622
(Huang et al., 2012)	**0.657**
(Neelakantan et al., 2014)	0.567
(Mu et al., 2017)	0.637
ours	0.652

Table 8: The results for different methods in the task of word similarity in context. The best result is in boldface. Our result is close to the best.

Note that Huang et al. (2012) retrained the vectors for the senses on the corpus, while our method depends only on senses extracted from the off-the-shelf vectors. After all, our goal is to show word senses already reside within off-the-shelf word vectors.

6.3 Police Lineup

Evaluating WSI systems can run into well-known difficulties, as reflected in the changing metrics over the years (Navigli and Vannella, 2013). Inspired by word-intrusion tests for topic coherence (Chang et al., 2009), we proposed a new simple test, which has the advantages of being easy to understand, and capable of being administered to humans.

The testbed uses 200 polysemous words and their 704 senses according to WordNet. Each sense is represented by 8 related words, which were collected from WordNet and online dictionaries by college students, who were told to identify most relevant other words occurring in the online definitions of this word sense as well as in the accompanying illustrative sentences. These are considered as ground truth representation of the word sense. These 8 words are typically not synonyms. For example, for the *tool/weapon* sense of *axe* they were "handle, harvest, cutting, split, tool, wood, battle, chop."

The quantitative test is called *police lineup*. First, randomly pick one of these 200 polysemous words. Second, pick the true senses for the word and then add randomly picked senses from other words so that there are n senses in total, where each sense is represented by 8 related words as mentioned. Finally, the algorithm (or human) is given the polysemous word and a set of n senses, and has to identify the true senses in this set. Table 9 gives an example.

word		senses
bat	1	**navigate nocturnal mouse wing cave sonic fly dark**
	2	**used hitting ball game match cricket play baseball**
	3	**wink briefly shut eyes wink bate quickly action**
	4	whereby legal court law lawyer suit bill judge
	5	loose ends two loops shoelaces tie rope string
	6	horny projecting bird oral nest horn hard food

Table 9: An example of the police lineup test with $n = 6$. The algorithm (or human subject) is given the polysemous word "bat" and $n = 6$ senses each of which is represented as a list of words, and is asked to identify the true senses belonging to "bat" (highlighted in boldface for demonstration).

Algorithm 1 Our method for the police lineup test

```
Input: Word $w$, list $S$ of senses (each has $8$ words)
Output: $t$ senses out of $S$
Heuristically find inflectional forms of $w$.
Find 5 atoms for $w$ and each inflectional form. Let $U$ denote the union of all these atoms.
Initialize the set of candidate senses $C_w \leftarrow \emptyset$, and the score for each sense $L$ to $\mathrm{score}(L) \leftarrow -\infty$
for each atom $a \in U$ do
    Rank senses $L \in S$ by
        $\mathrm{score}(a, L) = s(a, L) - s_A^L + s(w, L) - s_V^L$
    Add the two senses $L$ with highest $\mathrm{score}(a, L)$ to $C_w$, and update their scores
        $\mathrm{score}(L) \leftarrow \max\{\mathrm{score}(L), \mathrm{score}(a, L)\}$
Return the $t$ senses $L \in C_s$ with highest $\mathrm{score}(L)$
```

Our method (Algorithm 1) uses the similarities between any word (or atom) x and a set of words Y, defined as $s(x, Y) = \langle v_x, v_Y \rangle$ where v_Y is the SIF embedding of Y. It also uses the average similarities:

$$s_A^Y = \frac{\sum_{a \in A} s(a, Y)}{|A|}, \quad s_V^Y = \frac{\sum_{w \in V} s(w, Y)}{|V|}$$

where A are all the atoms, and V are all the words. We note two important practical details. First, while we have been using atoms of discourse as a proxy for word sense, these are too coarse-grained: the total number of senses (e.g., WordNet synsets) is far greater than 2000. Thus the score$(\cdot)$ function uses both the atom and the word vector. Second, some words are more popular than the others—i.e., have large components along many atoms and words—which seems to be an instance of the smoothing

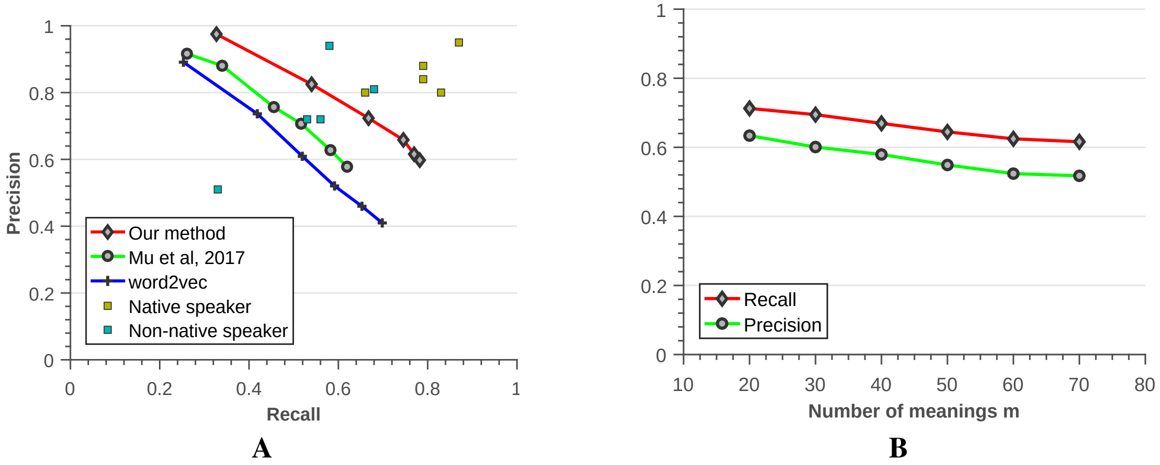

Figure 1: Precision and recall in the police lineup test. (**A**) For each polysemous word, a set of $n = 20$ senses containing the ground truth senses of the word are presented. Human subjects are told that on average each word has 3.5 senses and were asked to choose the senses they thought were true. The algorithms select t senses for $t = 1, 2, \ldots, 6$. For each t, each algorithm was run 5 times (standard deviations over the runs are too small to plot). (**B**) The performance of our method for $t = 4$ and $n = 20, 30, \ldots, 70$.

phenomenon alluded to in Footnote 4. The penalty terms s_A^L and s_V^L lower the scores of senses L containing such words. Finally, our algorithm returns t senses where t can be varied.

Results. The precision and recall for different n and t (number of senses the algorithm returns) are presented in Figure 1. Our algorithm outperforms the two selected competitors. For $n = 20$ and $t = 4$, our algorithm succeeds with precision 65% and recall 75%, and performance remains reasonable for $n = 50$. Giving the same test to humans[5] for $n = 20$ (see the left figure) suggests that our method performs similarly to non-native speakers.

Other word embeddings can also be used in the test and achieved slightly lower performance. For $n = 20$ and $t = 4$, the precision/recall are lower by the following amounts: GloVe $2.3\%/5.76\%$, NNSE (matrix factorization on PMI to rank 300 by Murphy et al. (2012)) $25\%/28\%$.

7 Conclusions

Different senses of polysemous words have been shown to lie in linear superposition inside standard word embeddings like word2vec and GloVe. This has also been shown theoretically building upon previous generative models, and empirical tests of this theory were presented. A priori, one imagines that showing such theoretical results about the inner structure of modern word embeddings would be hopeless since they are solutions to complicated nonconvex optimization.

A new WSI method is also proposed based upon these insights that uses only the word embeddings and sparse coding, and shown to provide very competitive performance on some WSI benchmarks. One novel aspect of our approach is that the word senses are interrelated using one of about 2000 discourse vectors that give a succinct description of which other words appear in the neighborhood with that sense. Our method based on sparse coding can be seen as a linear algebraic analog of the clustering approaches, and also gives fine-grained thematic structure reminiscent of topic models.

A novel police lineup test was also proposed for testing such WSI methods, where the algorithm is given a word w and word clusters, some of which belong to senses of w and the others are distractors belonging to senses of other words. The algorithm has to identify the ones belonging to w. We conjecture this police lineup test with distractors will challenge some existing WSI methods, whereas our method was found to achieve performance similar to non-native speakers.

[5] Human subjects are graduate students from science or engineering majors at major U.S. universities. Non-native speakers have 7 to 10 years of English language use/learning.

Acknowledgements

We thank the reviewers and the action editor of TACL for helpful feedback and thank the editors for granting special relaxation of the page limit for our paper. This work was supported in part by NSF grants CCF-1527371, DMS-1317308, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329. Tengyu Ma was additionally supported by the Simons Award in Theoretical Computer Science and by the IBM Ph.D. Fellowship.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transaction of Association for Computational Linguistics*, pages 385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *In Proceedings of International Conference on Learning Representations*.

Javier Artiles, Enrique Amigó, and Julio Gonzalo. 2009. The role of named entities in web people search. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 534–542.

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

David M. Blei. 2012. Probabilistic topic models. *Communication of the Association for Computing Machinery*, pages 77–84.

Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 288–296.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, pages 22–29.

Ivan Damnjanovic, Matthew Davies, and Mark Plumbley. 2010. SMALLbox – an evaluation framework for sparse representations and dictionary learning algorithms. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 418–425.

Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, pages 709–754.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of Association for Computational Linguistics*, pages 1491–1500.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

John Rupert Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 2017. Skip-gram – Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 69–76.

Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological review*, pages 211–244.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. 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*, pages 873–882.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning sense embeddings for word and relational similarity. In *Proceedings of Association for Computational Linguistics*, pages 95–105.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Suresh Manandhar, Ioannis P Klapaftis, Dmitriy Dligach, and Sameer S Pradhan. 2010. SemEval 2010: Task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68.

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*, pages 3111–3119.

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

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In

Proceedings of the 24th International Conference on Machine Learning, pages 641–648.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. Geometry of polysemy. *In Proceedings of International Conference on Learning Representations*.

Brian Murphy, Partha Pratim Talukdar, and Tom M. Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1933–1950.

Roberto Navigli and Daniele Vannella. 2013. SemEval 2013: Task 11: Word sense induction and disambiguation within an end-user application. In *Second Joint Conference on Lexical and Computational Semantics*, pages 193–201.

Arvind Neelakantan, Jeevan Shankar, Re Passos, and Andrew Mccallum. 2014. Efficient nonparametric estimation of multiple embeddings per word in vector space. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069.

Bruno Olshausen and David Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, pages 3311–3325.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing*, pages 1532–1543.

Joseph Reisinger and Raymond Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 107–117.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning*, pages 410–420.

Hinrich Schutze. 1998. Automatic word sense discrimination. *Computational Linguistics*, pages 97–123.

Ran Tian, Naoaki Okazaki, and Kentaro Inui. 2017. The mechanism of additive composition. *Machine Learning*, 106(7):1083–1130.

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

Wikimedia. 2012. English Wikipedia dump. Accessed March 2015.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, pages 189–196.