Integrating WordNet for Multiple Sense Embeddings in Vector Semantics

David Foley
Kutztown University of Pennsylvania
dfole581@live.kutztown.edu

Jugal Kalita
University of Colorado, Colorado Springs
jkalita@uccs.edu

Abstract

Popular distributional approaches to semantics allow for only a single embedding of any particular word. A single embedding per word conflates the distinct meanings of the word and their appropriate contexts, irrespective of whether those usages are related or completely disjoint. We compare models that use the graph structure of the knowledge base WordNet as a post-processing step to improve vector-space models with multiple sense embeddings for each word, and explore the application to word sense disambiguation.

Keywords: Vector Semantics, WordNet, Synonym Selection, Word Sense Disambiguation

1 INTRODUCTION

Vector semantics is a computational model of written language that encodes the usage of words in a vector space, which facilitates performing mathematical manipulations on words as vectors (Martin and Jurafsky, 2016; Turney and Pantel, 2010). These vectors encode the contexts of words across a corpus, and are learned based on word distributions throughout the text. Vectors can then be compared by various distance metrics, usually the cosine function, to determine the similarity of the underlying words. They also seem to possess some modest degree of compositionality, in the sense that the addition and subtraction of vectors can sometimes result in equations that appear to reflect semantically meaningful relationships between words (Mikolov et al., 2013a; Mikolov et al., 2013b). Because it allows for the use of these well studied techniques from linear algebra to be brought to bear on the difficult domain of semantics, vector space models (VSMs) have been the focus of much recent research in NLP.

While vector representations of word meaning are capable of capturing important semantic features of words and performing tasks like meaning comparison and analogizing, one of their shortcomings is their implicit assumption that a single written word type has exactly one meaning (or distribution) in a language. But many words clearly have different senses corresponding to distinct appropriate contexts. Building distributional vector space models that account for this polysemous behavior would allow for better performance on tasks involving context-sensitive words, most obviously word sense disambiguation. Previous research that attempted to resolve this issue is discussed at length in the next section. Most common methods either use clustering or introduce knowledge from an ontology. The goal of the present research is to develop or improve upon methods that take advantage of the semantic groups and relations codified in WordNet, and specifically to focus on the downstream WSD task, which is often neglected in favor of less useful similarity judgment evaluations.

The algorithm we examine in depth can in principle be implemented with any ontology, but in the present paper we focus exclusively on WordNet. WordNet (WN) is a knowledge base for English language semantics (Miller, 1995). It consists of small collections of synonymous words called synsets, interconnected with labeled links corresponding to different forms of semantic or lexical relations. We will be particularly interested in the synset relation of hypernymy/hyponymy. Hyponyms can be thought of as semantic subsets: If A is a hyponym of B, then x is A implies x is B, but the converse is not true. WordNet is also equipped with a dictionary definition for each synset, along with example sentences featuring varying synonymous words. Often implementations that use WordNet's graph structure fail to make use of these other features, which we will

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 2–9,
Varanasi, India. December 2016.

show can improve performance on several tasks.

2 Related Work

Our work is based primarily on that of Jauhar et al's RETROFIT algorithm (Jauhar et al., 2015), which is discussed at greater length in Section 3. Below we discuss previous models for building sense embeddings.

2.1 Clustering-Based Methods

(Reisinger and Mooney, 2010) learn a fixed number of sense vectors per word by clustering context vectors corresponding to individual occurrences of a word in a large corpus, then calculating the cluster centroids. These centroids are the sense vectors. (Huang et al., 2012) build a similar model using k-means clustering, but also incorporate global textual features into initial context vectors. They compile the Stanford Contextual Word Similarity dataset (SCWS), which consists of over two thousand word pairs in their sentential context, along with a similarity score based on human judgments from zero to ten. (Neelakantan et al., 2015) introduce an unsupervised modification of the skip-gram model (Mikolov et al., 2013b) to calculate multiple sense embeddings online, by maintaining clusters of context vectors and forming new word sense vectors when a context under consideration is sufficiently far from any of the word's known clusters. The advantage of the method is that it is capable of detecting different numbers of senses for different words, unlike the previous implementations of Huang et al. and Reisinger and Mooney.

2.2 Ontology-Based Methods

(Chen et al., 2014) first learn general word embeddings from the skip-gram model, then initialize sense embeddings based on the synsets and glosses of WN. These embeddings are then used to identify relevant occurrences of each sense in a training corpus using simple-to-complex words-sense disambiguation (S2C WSD). The skip-gram model is then trained directly on the disambiguated corpus. (Rothe and Schütze, 2015) build a neural-network post-processing system called AutoExtend that takes word embeddings and learns embeddings for synsets and lexemes. Their model is an autoencoder neural net with lexeme and synset embeddings as hidden layers, based on the intuition that a word is the sum of its lexemes and a synset is the sum of its lexemes. [3]

Our intuitions are most similar to those of (Jauhar et al., 2015) and we will be building on one of their approaches. Their RETROFIT algorithm learns embeddings for different word senses from WN by iteratively combining general embeddings according to the graph structure of WN. The approach is discussed in more detail below.

3 Improved Sense Embeddings from Word Embeddings

3.1 RETROFIT Algorithm

Because our work follows so directly from (Jauhar et al., 2015), we repeat the essential details of the RETROFIT algorithm here. Let $\Omega = (S_\Omega, E_\Omega)$ be a directed graph. We call Ω an *ontology* when the set of vertices S_Ω represent semantic objects of some kind and the set of edges E_Ω represent relationships between those objects. In the case of WN, S_Ω is the set of synsets and E_Ω are the semantic links (notably hypernyms and hyponyms). Given a set of sense-agnostic word embeddings $\hat{V}$ and an ontology Ω, RETROFIT infers a set of sense embeddings $\hat{S}$ that is maximally "consistent" with both $\hat{V}$ and Ω. By "consistency" we refer to the minimization of the objective function

$$\begin{aligned} D(\hat{S}) = &\sum_{ij} \alpha \left\| \hat{w}_i - \vec{s}_{ij} \right\|^2 \\ &+ \sum_{ij} \sum_{i'j' \in N_{ij}} \beta_r \left\| \vec{s}_{ij} - \vec{s}_{i'j'} \right\|^2 \end{aligned} \tag{1}$$

where s_{ij} is the *jth* sense of the *ith* word, N_{ij} is the set of neighbors of s_{ij} defined in E_Ω and α and β are hyperparameters controlling the importance of intial sense-agnositc embeddings and various ontological relationships, respectively. Essentially RETROFIT aims to make a sense embedding as similar to its sense-agnostic embedding as possible, while also reducing the distance between related senses as defined by Ω. It achieves this by iteratively updating sense embeddings according to

$$\vec{s}_{ij} = \frac{\alpha \hat{w}_i + \sum_{i'j' \in N_{ij}} \beta_r \vec{s}_{i'j'}}{\alpha + \sum_{i'j' \in N_{ij}} \beta_r} \tag{2}$$

until convergence. The RETROFIT implementation discussed in (Jauhar et al., 2015) defines only synonym, hypernym and hyponym relations, with respective weights of $\beta_r = 1.0, 0.5$ and 0.5

The RETROFIT algorithm genertes embeddings for word senses only from words whose surface form matches the entry in WordNet. Below we discuss several of the limitations associated with this RETROFIT implementation and possible improvements.

3.1.1 Impoverished Synsets

Many word senses are relatively isolated in the WordNet structure. They occur in synsets with few or no synonyms or semantic relations. In the case that the word has only one meaning, this is not a problem, because the sense-agnostic embedding is in that case unambiguous. But in the case that the word has one or more other semantically rich senses (ie, senses with synonyms and hyper/hyponym relations), the impoverished sense is unduly influenced by the general embedding and its unique meaning is not distinguishable. In the extreme case both senses are identical. Thousands of such synsets exist, including the synsets for words such as *inclement* and *egalitarian*.

3.1.2 Compound Words and Multi-word Lemmas

The original RETROFIT implementation discards multi-word lemmas (and entire synsets if they consist only of multi-word lemmas.) But there exist synsets for whom most or all of the related WN synsets contain only multi-word lemmas. See, for instance, the noun form of the word *unseen*, or the more extreme case of the synset *brass.n.01*, which has eleven distinct hypernym and hyponym relations, all but two of which are compound words for types of brass. Adjusting the RETROFIT algorithm to allow for embeddings of the multi-word lemmas that appear in WN would greatly reduce the number of impoverished synsets.

3.1.3 Underrepresented Senses

The general embedding produced by word2vec[1] (Mikolov et al., 2013a; Mikolov et al., 2013b) conflates all usages of a word. If a particular sense of a word is significantly less common than others, the word2vec embedding will not be a good representation of the sense. RETROFIT indiscriminately tries to minimize the distance from any particular sense and its word2vec embedding. Consider the usage of the word *tap* given by the synset *tap.v.11*, meaning "to pierce in order to draw liquid from." This usage occurs nowhere in the labelled SemCor corpus (Mihalcea, 1998), and is plausibly not well represented by the word2vec sense-agnostic embedding.

[1] https://code.google.com/archive/p/word2vec/

3.2 Modified RETROFIT Algorithm

For these reasons we make the following modifications to RETROFIT.

1) Regardless of the position of a word sense in WordNet, it will be equipped with a descriptive gloss that clarifies its usage. We incorporate all content words from each synset's gloss in the RETROFIT algorithm's objective function, where "content words" refers to any word for which we have a sense-agnostic embedding. Content words that appear more than once in the gloss are weighted according to the number of times they occur (ie, if a word is repeated in the gloss, it has a stronger influence on the sense embedding.)

2) We implement a naive model to handle a compound word by simply representing its sense-agnostic embedding as the average of the sense-agnostic embeddings of its constituent words. Although this is obviously inadequate for many compound words, we find it is already an improvement.

3) The sense-agnostic embedding of a word is assumed to be the weighted average of its sense embeddings, proportional to how common a particular word sense is. We calculate the sense-frequencies from the SemCor corpus, which consists of around 300,000 words tagged with their WordNet 3.0 synsets (Mihalcea, 1998).

3.3 Weighted RETROFIT Algorithm

Weigthed RETROFIT proceeds very similarly to RETROFIT algorithm by (Jauhar et al., 2015). We begin by intializing an embedding for each word sense as the sense-agnostic embedding (or, in the case of multi-word lemmas, the average of the sense-agnostic embeddings of the constituant words). The embeddings are then iteratively updated to make them more similar to their semantic neighbors in the WordNet ontology, and to make the weighted average of the sense embeddings of a word closer to the sense-agnostic embedding. The weighted average is learned from the SemCor counts as discussed.

More precisely, let $M = (V, \hat{V}, S, \hat{S}, P, \Omega)$ be a model consisting of a vocabulary V and sense-agnostic embeddings $\hat{V}$, a set of word senses S and sense-embeddings $\hat{S}$, a discrete probability

density function $P : V \times S \rightarrow \mathbb{R}$, and an ontology Ω. We seek the set $\hat{S}$ that minimizes the new objective function for the weighted RETROFIT algorithm (Equation 3).

$$
\begin{aligned}
D(M) = & \sum_i \alpha \left\| \hat{w}_i - \textstyle\sum_j p_{ij} \vec{s}_{ij} \right\|^2 \\
& + \sum_{ij} \sum_{i'j' \in N_{ij}} \beta_r \left\| \vec{s}_{ij} - \vec{s}_{i'j'} \right\|^2 \\
& + \sum_{ij} \sum_{i' \in G_{ij}} \gamma \left\| \hat{w}_{i'} - \vec{s}_{ij} \right\|^2
\end{aligned} \tag{3}
$$

by iteratively updating embeddings according to Equation (4). where $\hat{w}_i \in \hat{V}$, $\vec{s}_{ij} \in \hat{S}$, $p_{ij} = P(s_{ij}|w_i)$, N_{ij} is the set of neighbor indices of the *jth* sense of the *ith* word defined in Ω, $G_{ij} = \{i : w_i \in \hat{V}$ is in the gloss of $s_{ij}\}$ and α, β_r and γ are the parameters controlling the weights of sense-agnostic word embeddings, relations and gloss words respectively. Note that iteratively updating the sense embeddings via Eqs. 2 or 4 is equivalent to optimizing their respective objective functions via coordinate descent.

4 Evaluation

We train three variations of the RETROFIT algorithm on the 50-dimensional global context vectors produced by (Huang et al., 2012): the unmodified RETROFIT, RETROFIT with gloss words and multi-word lemmas (which we refer to as Modified RETROFIT), and Weighted RETROFIT with weighted senses as discussed above. Training time is similar between the first two; weighted RETROFIT takes about twice as long. All converge to a solution within 0.01 within fifteen iterations.

The models are evaluated on two different tasks: Synonym Selection and Word Sense Disambiguation. We first include and discuss results from some similarity judgment tasks, but these serve more as stepping stone than an as a rigorous measure of model quality. (Faruqui et al., 2016) give a comprehensive assessment of the inadequacies of evaluating the quality of embeddings on word similarity tasks. In general, these tasks are fairly subjective and a model's performance on them does not correlate with performance on downstream NLP tasks.

4.1 Similarity Judgments

We evaluate the models on the RG-65 dataset, (Rubenstein and Goodenough, 1965) which consists of sixty-five pairs of words and an average human judgment of similarity scaled from one to four. Evaluation is a straightforward calculation of the average cosine similarity of each pair of sense embeddings, as used by (Jauhar et al., 2015) and originally proposed by (Reisinger and Mooney, 2010). As an exploration, we also consider the results of using the maximum cosine similarity, which returns the highest cosine similarity among any pair of senses from the respective words.

Our results are displayed in Table 1. Every model performs best on the task using the maximum cosine similarity metric, with our improved systems performing noticeably better. Interestingly, the commonly used average similarity metric causes our models to lose their advantage, particularly Weighted RETROFIT, whose chief credit is its ability to produce more distinct sense embeddings. Averaging these vectors together throws away the improvements gained by separating out the distinct meanings.

4.2 Synonym Selection

We test the models on two synonym selection datasets: ESL-50 (Turney, 2002) and TOEFL (Landauer and Dumais, 1997). ESL-50 is a set of fifty English sentences with a target word for which a synonym must be selected from four candidate words. TOEFL consists of eighty context-independent words and four potential candidates for each. For both datasets, we use the same maxSim selection criteria as (Jauhar et al., 2015). We select the sense vector $\vec{s}_{ij}$ that corresponds to:

$$maxSim(w_i, w_{i'}) = \max_{j,j'} cos(\vec{s}_{ij}, \vec{s}_{i'j'})$$

Our results are presented in Table 2. The results on this task are less straightforward. Although the ESL-50 and TOEFL datasets are remarkably similar in form, the models do not perform consistently across them. Our modified RETROFIT method produces an enormous improvement on TOEFL, while ESL-50 gives our models some difficulties. Whether this is an effect of the relatively small number of words in the task or whether there are specific features about how the datasets were assembled is unclear.

$$\bar{s}_{ij} = \frac{\alpha p_{ij}\hat{w}_i - \alpha p_{ij}\sum_{k\neq j} p_{ik}\vec{s}_{ik} + \sum_{i'j'\in N_{ij}} \beta_r \vec{s}_{i'j'} + \gamma \sum_{i'\in G_{ij}} \hat{w}_{i'}}{\alpha p_{ij}^2 + \sum_{i'j'\in N_{ij}} \beta_r + \sum_{i'\in G_{ij}} \gamma} \tag{4}$$

Similarity Judgments		
	RG-65	
	AVG	MAX
RETROFIT	**0.73**	0.79
Modified RETROFIT	0.72	**0.85**
Weighted RETROFIT	0.69	0.84

Table 1: Performance on RG-65 word similarity dataset. Scores are Spearman's rank correlation.

Synonym Selection		
	ESL-50	TOEFL
RETROFIT	**64.0**	68.75
Modified RETROFIT	62.0	**81.25**
Weighted RETROFIT	60.0	75.0

Table 2: Percent accuracy on ESL-50 and TOEFL synonym selection using maxSim comparison

4.3 Word Sense Disambiguation

We use Semeval 2015 task 13 (Moro and Navigli, 2015) as our English WSD test. The corpus for the task consists of four documents taken from the biomedical, mathematical and social issues domains, annotated with part of speech information. The task also includes named entity disambiguation, which we do not handle, except in the incidental case where there is a WN synset for a named entity. We explore two different methods for WSD. The first chooses a word sense by identifying a word that co-occurs in the sentence and has a sense that is closest to a sense of our target word. The intuition of the model is that although particular words may be totally unrelated to the sense of the target word, there should exist somewhere in the sentence a word pertaining to the subject described by the ambiguous word. Formally, this method is described as the *contextMax* function:

$$contextMax(w, c) = \arg\max_{s\in S_i}\Big(\max_{c\in \bigcup_{k\neq i} S_k} cos(\vec{s}, \vec{c}) \cdot p(s|w)\Big) \tag{5}$$

where S_i is the set of senses of the *ith* word of the context sentence.

The second WSD method incorporates both local and global context in equal parts. The intuition is that nearby words in a particular sentence will capture information about the particular usage of a word, while words that appear over the course of a passage will characterize the subject matter being discussed. Both of these component are essential to human understanding and should aid WSD algorithms, as discussed in (Weissenborn et al., 2015). Formally, we define the localGlobal WSD function as

$$localGlobal(w, c) = \arg\max_{s\in W_{ij}}(cos(\vec{s}, \vec{c}_{ij})\cdot p(s|w)) \tag{6}$$

where the context vector $\vec{c}_{ij}$ for the *jth* word of the *ith* sentence is given by

$$\vec{c}_{ij} = \frac{\vec{l}_{ij}}{|\vec{l}_{ij}|} + \frac{\vec{g}_i}{|\vec{g}_i|}$$

and the local context vector $\vec{l}_{ij}$ of the *jth* word of the *ith* sentence and global context vector $\vec{g}_i$ of the *ith* sentence are given by

$$\vec{l}_{ij} = \sum_{k\neq j} \frac{1}{|j-k|}\hat{w}_{ik}$$

$$\vec{g}_i = \sum_{n=i-2}^{i+2}\sum_k \hat{w}_{nk}$$

As a baseline we compare against the most-frequent sense tagger (MFS) trained on the Semcor corpus (Moro and Navigli, 2015), defined simply as

$$mfs(w) = \arg\max_{s\in S_w}(p(s|w)) \tag{7}$$

Word Sense Disambiguation					
	Nouns	Verbs	Adjectives	Adverbs	All
MFS	45.8	49.9	67.5	70.6	53.5
RETROFIT	49.1	52.0	67.3	75.3	56.2
Modified RETROFIT	**50.6**	50.0	**69.2**	**76.5**	**57.0**
Weighted RETROFIT	50.0	**52.8**	65.4	**76.5**	56.8

Table 3: Semeval 2015 task 13 F1 scores of the models using the contextMax disambiguation function.

	Nouns	Verbs	Adjectives	Adverbs	All
RETROFIT	52.5	57.2	**77.3**	77.8	61.1
Modified RETROFIT	53.6	56.4	76.0	**79.0**	61.6
Weighted RETROFIT	**53.9**	**59.2**	75.4	77.8	**62.1**

Table 4: Semeval 2015 task 13 F1 scores of the models using the contextMax disambiguation function, restricted to correct POS

Tables 3 and 4 display results for our models using contextMax disambiguation with and without restriction by POS information, along with the MFS tagging baseline. In both cases, RETROFIT and MFS are outperformed overall by our improvements. Tables 5 and 6 show the WSD results using localGlobal disambiguation, which for the most part appears to be a strictly better metric. Results are ranked by F1 score, the harmonic mean of precision and recall (uniformly weighted). Although it underperforms on the compartively easier task of disambiguating adjectivs and adverbs, Weighted RETROFIT is the best model of verbs by every single metric.

By all measures, the various RETROFIT implementations outperform the MFS baseline. Weighted RETROFIT and Modified RETROFIT both improve the initial model. The best performing systems on the Semeval 2015 task 13 English corpus are LIMSI and SUDOKU (Moro and Navigli, 2015), which achieve F1 scores of 65.8 and 61.6 respectively. This would position both Weighted RETROFIT and RETROFIT with compound words and gloss words as second only to the top system, even with the use of relatively low dimensional embeddings.

5 Discussion

Results on similarity judgment are mixed, although it should be noted that despite the fact that in principle average similarity appears to be a good measure of word relatedness, in our trials the maximum similarity between two words is a better predictor of human judgments on RG-65 with all algorithms. It's possible that in the absence of disambiguating context human judges are not actually good at combining the relatedness of different senses of words and instead specifically search for related meanings when evaluating similarity. It's worth noting that the metric by which our modifications provide the largest improvements is the metric which RETROFIT itself also performs best by. But, as discussed above and in [4], even human judges often do not score particularly well similarity tasks, and in fact there may be no real "gold standard" on such a task.

The results of the synonym selection task are also mixed. On the ESL-50 dataset our modifications slightly underperform, while on the TOEFL dataset they provide an enormous improvement. We have not investigated the particulars of the datasets enough to see if there are anomolous features (over or under-representation of certain parts of speech, rare word senses, etc), or if these performance gaps are due more to the small sample size of the test data. Testing on a wider array of larger synonym selection datasets could yield insight into the models' shortcomings.

Our models are a noticeable improvement on WSD. Interestingly, the Weighted RETROFIT algorithm achieves the best scores on verbs across

	Nouns	Verbs	Adjectives	Adverbs	All
RETROFIT	49.5	49.2	64.2	**79.0**	55.7
Modified RETROFIT	**54.8**	50.0	**67.9**	77.8	**59.5**
Weighted RETROFIT	53.0	**52.4**	62.3	74.1	57.9

Table 5: Semeval 2015 task 13 F1 scores of the models using the localGlobal disambiguation function

	Nouns	Verbs	Adjectives	Adverbs	All
RETROFIT	52.2	55.6	73.5	**80.2**	60.2
Modified RETROFIT	**56.6**	57.6	**74.1**	**80.2**	**63.4**
Weighted RETROFIT	55.6	**59.2**	72.9	76.5	62.1

Table 6: Semeval 2015 task 13 F1 scores of the models using the localGlobal disambiguation function, restricted to correct POS

all metrics. Again, whether this is a quirk of the specific corpus is unclear. If not, it may indicate that homophonous verbs in English tend to be more distinct from each other than other parts of speech, perhaps because of more common metaphorical language use. We at least can say confidently that utilizing more features from WN is an across the board improvement.

Future Work

As mentioned above, the limited size and scope of the test sets leaves room for doubt about the models' performance on new datasets, especially when two datasets for the same task yield strikingly different results, like synonym selection. A useful exploration may be looking at domain-specific datasets for this task, as the results might suggest that the performance discrepancies are present between domains. It is possible, for example, that WordNet underrepresents certain domains. (Consider the case of the word *nugget*, which in WordNet has no synsets related to food, but in American English is most often used in the compound *chicken nugget*.) It will also be important to try the same task with significantly larger datasets.

We also use only a crude model of compound word vectors. An investigation of better compositional semantic models could greatly benefit the algorithm, as a large percentage of WN synsets contain compound words.

The RETROFIT algorithm may also be discarding valuable information by constructing the sense vectors only from the sense-agnostic embeddings for words whose exact surface form matches entries in WordNet. But word2vec and most other VSM algorithms learn embeddings for many different conjugations of words, and in fact those conjugations may themselves contain information (such as part-of-speech) that can help further differentiate senses.

Our models are all trained on the relatively low dimensional global feature vectors produced by (Huang et al., 2012), but significantly richer embeddings exist, such as the GoogleNews vectors, which are 300 dimensional and were trained on a 100 billion word corpus using CBOW (Mikolov et al., 2013a; Mikolov et al., 2013b). We expect that the quality of the embeddings produced by the RETROFIT algorithms will scale with the quality of the underlying embeddings, and can hope for continual improvement as larger and better datasets become available.

6 Acknowledgement

This work was funded under NSF grant 1359275. The authors would also like to thank Feras Al Tarouti for his valuable input.

References

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*, pages 1025–1035.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of

word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

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: Long Papers-Volume 1*, pages 873–882.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of National Association for Computational Linguistics (NAACL)*.

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.

James H Martin and Daniel Jurafsky. 2016. Semantics with dense vectors, in speech and language processing. *Third Edition, Draft*.

Rada Mihalcea. 1998. SemCor semantically tagged corpus. *Unpublished manuscript*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Peter D Turney. 2002. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *arXiv preprint cs/0212033*.

Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. 2015. Multi-objective optimization for the joint disambiguation of nouns and named entities. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 596–605.