Dependency-Based Word Embeddings

Omer Levy* and **Yoav Goldberg**
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{omerlevy,yoav.goldberg}@gmail.com

Abstract

While continuous word embeddings are gaining popularity, current models are based solely on linear contexts. In this work, we generalize the skip-gram model with negative sampling introduced by Mikolov et al. to include arbitrary contexts. In particular, we perform experiments with dependency-based contexts, and show that they produce markedly different embeddings. The dependency-based embeddings are less topical and exhibit more functional similarity than the original skip-gram embeddings.

1 Introduction

Word representation is central to natural language processing. The default approach of representing words as discrete and distinct symbols is insufficient for many tasks, and suffers from poor generalization. For example, the symbolic representation of the words "pizza" and "hamburger" are completely unrelated: even if we know that the word "pizza" is a good argument for the verb "eat", we cannot infer that "hamburger" is also a good argument. We thus seek a representation that captures semantic and syntactic similarities between words. A very common paradigm for acquiring such representations is based on the distributional hypothesis of Harris (1954), stating that words in similar contexts have similar meanings.

Based on the distributional hypothesis, many methods of deriving word representations were explored in the NLP community. On one end of the spectrum, words are grouped into clusters based on their contexts (Brown et al., 1992; Uszkoreit and Brants, 2008). On the other end, words are represented as a very high dimensional but sparse vectors in which each entry is a measure of the association between the word and a particular context (see (Turney and Pantel, 2010; Baroni and Lenci, 2010) for a comprehensive survey). In some works, the dimensionality of the sparse word-context vectors is reduced, using techniques such as SVD (Bullinaria and Levy, 2007) or LDA (Ritter et al., 2010; Séaghdha, 2010; Cohen et al., 2012). Most recently, it has been proposed to represent words as dense vectors that are derived by various training methods inspired from neural-network language modeling (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2008; Mikolov et al., 2011; Mikolov et al., 2013b). These representations, referred to as "neural embeddings" or "word embeddings", have been shown to perform well across a variety of tasks (Turian et al., 2010; Collobert et al., 2011; Socher et al., 2011; Al-Rfou et al., 2013).

Word embeddings are easy to work with because they enable efficient computation of word similarities through low-dimensional matrix operations. Among the state-of-the-art word-embedding methods is the *skip-gram with negative sampling* model (SKIPGRAM), introduced by Mikolov et al. (2013b) and implemented in the `word2vec` software.[1] Not only does it produce useful word representations, but it is also very efficient to train, works in an online fashion, and scales well to huge copora (billions of words) as well as very large word and context vocabularies.

Previous work on neural word embeddings take the contexts of a word to be its *linear context* – words that precede and follow the target word, typically in a window of k tokens to each side. However, other types of contexts can be explored too.

In this work, we generalize the SKIPGRAM model, and move from linear bag-of-words contexts to arbitrary word contexts. Specifically,

* Supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

[1] `code.google.com/p/word2vec/`

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 302–308,
Baltimore, Maryland, USA, June 23-25 2014.

following work in sparse vector-space models (Lin, 1998; Padó and Lapata, 2007; Baroni and Lenci, 2010), we experiment with *syntactic contexts* that are derived from automatically produced dependency parse-trees.

The different kinds of contexts produce noticeably different embeddings, and induce different word similarities. In particular, the bag-of-words nature of the contexts in the "original" SKIPGRAM model yield *broad topical similarities*, while the dependency-based contexts yield more *functional* similarities of a *cohyponym* nature. This effect is demonstrated using both qualitative and quantitative analysis (Section 4).

The neural word-embeddings are considered opaque, in the sense that it is hard to assign meanings to the dimensions of the induced representation. In Section 5 we show that the SKIPGRAM model does allow for some introspection by querying it for contexts that are "activated by" a target word. This allows us to peek into the learned representation and explore the contexts that are found by the learning process to be most discriminative of particular words (or groups of words). To the best of our knowledge, this is the first work to suggest such an analysis of discriminatively-trained word-embedding models.

2 The Skip-Gram Model

Our departure point is the skip-gram neural embedding model introduced in (Mikolov et al., 2013a) trained using the negative-sampling procedure presented in (Mikolov et al., 2013b). In this section we summarize the model and training objective following the derivation presented by Goldberg and Levy (2014), and highlight the ease of incorporating arbitrary contexts in the model.

In the skip-gram model, each word $w \in W$ is associated with a vector $v_w \in R^d$ and similarly each context $c \in C$ is represented as a vector $v_c \in R^d$, where W is the words vocabulary, C is the contexts vocabulary, and d is the embedding dimensionality. The entries in the vectors are latent, and treated as parameters to be learned. Loosely speaking, we seek parameter values (that is, vector representations for both words and contexts) such that the dot product $v_w \cdot v_c$ associated with "good" word-context pairs is maximized.

More specifically, the negative-sampling objective assumes a dataset D of observed (w, c) pairs of words w and the contexts c, which appeared in a large body of text. Consider a word-context pair (w, c). Did this pair come from the data? We denote by $p(D = 1|w, c)$ the probability that (w, c) came from the data, and by $p(D = 0|w, c) = 1 - p(D = 1|w, c)$ the probability that (w, c) did not. The distribution is modeled as:

$$p(D = 1|w, c) = \frac{1}{1+e^{-v_w \cdot v_c}}$$

where v_w and v_c (each a d-dimensional vector) are the model parameters to be learned. We seek to maximize the log-probability of the observed pairs belonging to the data, leading to the objective:

$$\arg\max_{v_w, v_c} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c \cdot v_w}}$$

This objective admits a trivial solution in which $p(D = 1|w, c) = 1$ for every pair (w, c). This can be easily achieved by setting $v_c = v_w$ and $v_c \cdot v_w = K$ for all c, w, where K is large enough number.

In order to prevent the trivial solution, the objective is extended with (w, c) pairs for which $p(D = 1|w, c)$ must be low, i.e. pairs which are not in the data, by generating the set D' of random (w, c) pairs (assuming they are all incorrect), yielding the negative-sampling training objective:

$$\arg\max_{v_w, v_c} \left(\prod_{(w,c)\in D} p(D = 1|c, w) \prod_{(w,c)\in D'} p(D = 0|c, w) \right)$$

which can be rewritten as:

$$\arg\max_{v_w, v_c} \left(\sum_{(w,c)\in D} \log \sigma(v_c \cdot v_w) + \sum_{(w,c)\in D'} \log \sigma(-v_c \cdot v_w) \right)$$

where $\sigma(x) = 1/(1+e^x)$. The objective is trained in an online fashion using stochastic-gradient updates over the corpus $D \cup D'$.

The negative samples D' can be constructed in various ways. We follow the method proposed by Mikolov et al.: for each $(w, c) \in D$ we construct n samples $(w, c_1), \ldots, (w, c_n)$, where n is a hyperparameter and each c_j is drawn according to its unigram distribution raised to the $3/4$ power.

Optimizing this objective makes observed word-context pairs have similar embeddings, while scattering unobserved pairs. Intuitively, words that appear in similar contexts should have similar embeddings, though we have not yet found a formal proof that SKIPGRAM does indeed maximize the dot product of similar words.

3 Embedding with Arbitrary Contexts

In the SKIPGRAM embedding algorithm, the contexts of a word w are the words surrounding it

in the text. The context vocabulary C is thus identical to the word vocabulary W. However, this restriction is not required by the model; contexts need not correspond to words, and the number of context-types can be substantially larger than the number of word-types. We generalize SKIPGRAM by replacing the bag-of-words contexts with arbitrary contexts.

In this paper we experiment with dependency-based *syntactic contexts*. Syntactic contexts capture different information than bag-of-word contexts, as we demonstrate using the sentence "*Australian scientist discovers star with telescope*".

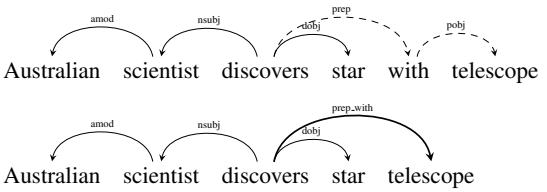
Linear Bag-of-Words Contexts This is the context used by `word2vec` and many other neural embeddings. Using a window of size k around the target word w, $2k$ contexts are produced: the k words before and the k words after w. For $k = 2$, the contexts of the target word w are $w_{-2}, w_{-1}, w_{+1}, w_{+2}$. In our example, the contexts of *discovers* are *Australian*, *scientist*, *star*, *with*.[2]

Note that a context window of size 2 may miss some important contexts (*telescope* is not a context of *discovers*), while including some accidental ones (*Australian* is a context *discovers*). Moreover, the contexts are unmarked, resulting in *discovers* being a context of both *stars* and *scientist*, which may result in *stars* and *scientists* ending up as neighbours in the embedded space. A window size of 5 is commonly used to capture broad topical content, whereas smaller windows contain more focused information about the target word.

Dependency-Based Contexts An alternative to the bag-of-words approach is to derive contexts based on the syntactic relations the word participates in. This is facilitated by recent advances in parsing technology (Goldberg and Nivre, 2012; Goldberg and Nivre, 2013) that allow parsing to syntactic dependencies with very high speed and near state-of-the-art accuracy.

After parsing each sentence, we derive word contexts as follows: for a target word w with modifiers $m_1, \ldots, m_k$ and a head h, we consider the contexts $(m_1, lbl_1), \ldots, (m_k, lbl_k), (h, lbl_h^{-1})$,

[2] `word2vec`'s implementation is slightly more complicated. The software defaults to prune rare words based on their frequency, and has an option for sub-sampling the frequent words. These pruning and sub-sampling happen *before* the context extraction, leading to a dynamic window size. In addition, the window size is not fixed to k but is sampled uniformly in the range $[1, k]$ for each word.



WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep_with^{-1}

Figure 1: Dependency-based context extraction example. **Top:** preposition relations are collapsed into single arcs, making *telescope* a direct modifier of *discovers*. **Bottom:** the contexts extracted for each word in the sentence.

where lbl is the type of the dependency relation between the head and the modifier (e.g. *nsubj*, *dobj*, *prep_with*, *amod*) and lbl^{-1} is used to mark the inverse-relation. Relations that include a preposition are "collapsed" prior to context extraction, by directly connecting the head and the object of the preposition, and subsuming the preposition itself into the dependency label. An example of the dependency context extraction is given in Figure 1.

Notice that syntactic dependencies are both more inclusive and more focused than bag-of-words. They capture relations to words that are far apart and thus "out-of-reach" with small window bag-of-words (e.g. the instrument of *discover* is *telescope/prep_with*), and also filter out "coincidental" contexts which are within the window but not directly related to the target word (e.g. *Australian* is not used as the context for *discovers*). In addition, the contexts are typed, indicating, for example, that *stars* are objects of discovery and *scientist*s are subjects. We thus expect the syntactic contexts to yield more focused embeddings, capturing more functional and less topical similarity.

4 Experiments and Evaluation

We experiment with 3 training conditions: BOW5 (bag-of-words contexts with $k = 5$), BOW2 (same, with $k = 2$) and DEPS (dependency-based syntactic contexts). We modified `word2vec` to support arbitrary contexts, and to output the context embeddings in addition to the word embeddings. For bag-of-words contexts we used the original `word2vec` implementation, and for syntactic contexts, we used our modified version. The negative-sampling parameter (how many negative contexts to sample for every correct one) was 15.

All embeddings were trained on English Wikipedia. For DEPS, the corpus was tagged with parts-of-speech using the Stanford tagger (Toutanova et al., 2003) and parsed into labeled Stanford dependencies (de Marneffe and Manning, 2008) using an implementation of the parser described in (Goldberg and Nivre, 2012). All tokens were converted to lowercase, and words and contexts that appeared less than 100 times were filtered. This resulted in a vocabulary of about 175,000 words, with over 900,000 distinct syntactic contexts. We report results for 300 dimension embeddings, though similar trends were also observed with 600 dimensions.

4.1 Qualitative Evaluation

Our first evaluation is qualitative: we manually inspect the 5 most similar words (by cosine similarity) to a given set of target words (Table 1).

The first target word, *Batman*, results in similar sets across the different setups. This is the case for many target words. However, other target words show clear differences between embeddings.

In *Hogwarts* - the school of magic from the fictional Harry Potter series - it is evident that BOW contexts reflect the *domain* aspect, whereas DEPS yield a list of famous schools, capturing the *semantic type* of the target word. This observation holds for *Turing*[3] and many other nouns as well; BOW find words that *associate* with w, while DEPS find words that *behave* like w. Turney (2012) described this distinction as *domain similarity* versus *functional similarity*.

The *Florida* example presents an ontological difference; bag-of-words contexts generate meronyms (counties or cities within Florida), while dependency-based contexts provide cohyponyms (other US states). We observed the same behavior with other geographical locations, particularly with countries (though not all of them).

The next two examples demonstrate that similarities induced from DEPS share a syntactic function (adjectives and gerunds), while similarities based on BOW are more diverse. Finally, we observe that while both BOW5 and BOW2 yield topical similarities, the larger window size result in more topicality, as expected.

[3] DEPS generated a list of scientists whose name ends with "ing". This is may be a result of occasional POS-tagging errors. Still, the embedding does a remarkable job and retrieves scientists, despite the noisy POS. The list contains more mathematicians without "ing" further down.

Target Word	BOW5	BOW2	DEPS
batman	nightwing	superman	superman
	aquaman	superboy	superboy
	catwoman	aquaman	supergirl
	superman	catwoman	catwoman
	manhunter	batgirl	aquaman
hogwarts	dumbledore	evernight	sunnydale
	hallows	sunnydale	collinwood
	half-blood	garderobe	calarts
	malfoy	blandings	greendale
	snape	collinwood	millfield
turing	nondeterministic	non-deterministic	pauling
	non-deterministic	finite-state	hotelling
	computability	nondeterministic	heting
	deterministic	buchi	lessing
	finite-state	primality	hamming
florida	gainesville	fla	texas
	fla	alabama	louisiana
	jacksonville	gainesville	georgia
	tampa	tallahassee	california
	lauderdale	texas	carolina
object-oriented	aspect-oriented	aspect-oriented	event-driven
	smalltalk	event-driven	domain-specific
	event-driven	objective-c	rule-based
	prolog	dataflow	data-driven
	domain-specific	4gl	human-centered
dancing	singing	singing	singing
	dance	dance	rapping
	dances	dances	breakdancing
	dancers	breakdancing	miming
	tap-dancing	clowning	busking

Table 1: Target words and their 5 most similar words, as induced by different embeddings.

We also tried using the subsampling option (Mikolov et al., 2013b) with BOW contexts (not shown). Since `word2vec` removes the subsampled words from the corpus *before* creating the window contexts, this option effectively increases the window size, resulting in greater topicality.

4.2 Quantitative Evaluation

We supplement the examples in Table 1 with quantitative evaluation to show that the qualitative differences pointed out in the previous section are indeed widespread. To that end, we use the WordSim353 dataset (Finkelstein et al., 2002; Agirre et al., 2009). This dataset contains pairs of similar words that reflect either *relatedness* (topical similarity) or *similarity* (functional similarity) relations.[4] We use the embeddings in a retrieval/ranking setup, where the task is to rank the *similar* pairs in the dataset above the *related* ones.

The pairs are ranked according to cosine similarities between the embedded words. We then draw a recall-precision curve that describes the embedding's affinity towards one subset ("similarity") over another ("relatedness"). We expect DEPS's curve to be higher than BOW2's curve, which in turn is expected to be higher than

[4] Some word pairs are judged to exhibit both types of similarity, and were ignored in this experiment.

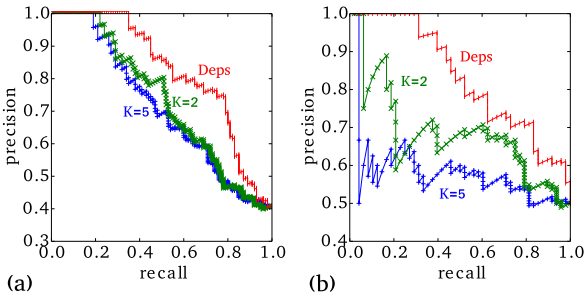


Figure 2: Recall-precision curve when attempting to rank the *similar* words above the *related* ones. (a) is based on the WordSim353 dataset, and (b) on the Chiarello et al. dataset.

batman	**hogwarts**	**turing**
superman/conj^{-1}	students/prep_at^{-1}	machine/nn^{-1}
spider-man/conj^{-1}	educated/prep_at^{-1}	test/nn^{-1}
superman/conj	student/prep_at^{-1}	theorem/poss^{-1}
spider-man/conj	stay/prep_at^{-1}	machines/nn^{-1}
robin/conj	learned/prep_at^{-1}	tests/nn^{-1}
florida	**object-oriented**	**dancing**
marlins/nn^{-1}	programming/amod^{-1}	dancing/conj
beach/appos^{-1}	language/amod^{-1}	dancing/conj^{-1}
jacksonville/appos^{-1}	framework/amod^{-1}	singing/conj^{-1}
tampa/appos^{-1}	interface/amod^{-1}	singing/conj
florida/conj^{-1}	software/amod^{-1}	ballroom/nn

Table 2: Words and their top syntactic contexts.

BOW5's. The graph in Figure 2a shows this is indeed the case. We repeated the experiment with a different dataset (Chiarello et al., 1990) that was used by Turney (2012) to distinguish between domain and functional similarities. The results show a similar trend (Figure 2b). When reversing the task such that the goal is to rank the *related* terms above the *similar* ones, the results are reversed, as expected (not shown).[5]

5 Model Introspection

Neural word embeddings are often considered opaque and uninterpretable, unlike sparse vector space representations in which each dimension corresponds to a particular known context, or LDA models where dimensions correspond to latent topics. While this is true to a large extent, we observe that SKIPGRAM does allow a non-trivial amount of introspection. Although we cannot assign a meaning to any particular dimension, we can indeed get a glimpse at the kind of information being captured by the model, by examining which contexts are "activated" by a target word.

Recall that the learning procedure is attempting to maximize the dot product $v_c \cdot v_w$ for good (w, c) pairs and minimize it for bad ones. If we keep the context embeddings, we can query the model for the contexts that are most activated by (have the highest dot product with) a given target word. By doing so, we can see what the model learned to be a good discriminative context for the word.

To demonstrate, we list the 5 most activated contexts for our example words with DEPS embeddings in Table 2. Interestingly, the most discriminative syntactic contexts in these cases are not associated with subjects or objects of verbs (or their inverse), but rather with conjunctions, appositions, noun-compounds and adjectivial modifiers. Additionally, the collapsed preposition relation is very useful (e.g. for capturing the *school* aspect of *hogwarts*). The presence of many conjunction contexts, such as `superman/conj` for *batman* and `singing/conj` for *dancing*, may explain the functional similarity observed in Section 4; conjunctions in natural language tend to enforce their conjuncts to share the same semantic types and inflections.

In the future, we hope that insights from such model introspection will allow us to develop better contexts, by focusing on conjunctions and prepositions for example, or by trying to figure out why the subject and object relations are absent and finding ways of increasing their contributions.

6 Conclusions

We presented a generalization of the SKIPGRAM embedding model in which the linear bag-of-words contexts are replaced with arbitrary ones, and experimented with dependency-based contexts, showing that they produce markedly different kinds of similarities. These results are expected, and follow similar findings in the distributional semantics literature. We also demonstrated how the resulting embedding model can be queried for the discriminative contexts for a given word, and observed that the learning procedure seems to favor relatively local syntactic contexts, as well as conjunctions and objects of preposition. We hope these insights will facilitate further research into improved context modeling and better, possibly task-specific, embedded representations. Our software, allowing for experimentation with arbitrary contexts, together with the embeddings described in this paper, are available for download at the authors' websites.

[5] Additional experiments (not presented in this paper) reinforce our conclusion. In particular, we found that DEPS perform dramatically worse than BOW contexts on analogy tasks as in (Mikolov et al., 2013c; Levy and Goldberg, 2014).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June. Association for Computational Linguistics.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proc. of CoNLL 2013*.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

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

Peter F Brown, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural. *Computational Linguistics*, 18(4).

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

Christine Chiarello, Curt Burgess, Lorie Richards, and Alma Pollock. 1990. Semantic and associative priming in the cerebral hemispheres: Some words do, some words don't... sometimes, some places. *Brain and Language*, 38(1):75–104.

Raphael Cohen, Yoav Goldberg, and Michael Elhadad. 2012. Domain adaptation of a dependency parser with a class-class selectional preference model. In *Proceedings of ACL 2012 Student Research Workshop*, pages 43–48, Jeju Island, Korea, July. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK, August. Coling 2008 Organizing Committee.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for the arc-eager system. In *Proc. of COLING 2012*.

Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *Transactions of the association for Computational Linguistics*, 1.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

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

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, pages 1081–1088.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *ACL*, pages 424–434.

Diarmuid Ó Séaghdha. 2010. Latent variable models of selectional preference. In *ACL*, pages 435–444.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Chris Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

P.D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *Proc. of ACL*, pages 755–762.