# Non-distributional Word Vector Representations

**Manaal Faruqi** and **Chris Dyer**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{mfaruqui, cdyer}@cs.cmu.edu

## Abstract

Data-driven representation learning for words is a technique of central importance in NLP. While indisputably useful as a source of features in downstream tasks, such vectors tend to consist of uninterpretable components whose relationship to the categories of traditional lexical semantic theories is tenuous at best. We present a method for constructing interpretable word vectors from hand-crafted linguistic resources like WordNet, FrameNet etc. These vectors are binary (i.e, contain only 0 and 1) and are 99.9% sparse. We analyze their performance on state-of-the-art evaluation methods for distributional models of word vectors and find they are competitive to standard distributional approaches.

## 1 Introduction

Distributed representations of words have been shown to benefit a diverse set of NLP tasks including syntactic parsing (Lazaridou et al., 2013; Bansal et al., 2014), named entity recognition (Guo et al., 2014) and sentiment analysis (Socher et al., 2013). Additionally, because they can be induced directly from unannotated corpora, they are likewise available in domains and languages where traditional linguistic resources do not exhaust. Intrinsic evaluations on various tasks are helping refine vector learning methods to discover representations that captures many facts about lexical semantics (Turney, 2001; Turney and Pantel, 2010).

Yet induced word vectors do not look anything like the representations described in most lexical semantic theories, which focus on identifying classes of words (Levin, 1993; Baker et al., 1998; Schuler, 2005; Miller, 1995). Though expensive to construct, conceptualizing word meanings symbolically is important for theoretical understanding and interpretability is desired in computational models.

Our contribution to this discussion is a new technique that constructs task-independent word vector representations using linguistic knowledge derived from pre-constructed linguistic resources like WordNet (Miller, 1995), FrameNet (Baker et al., 1998), Penn Treebank (Marcus et al., 1993) etc. In such word vectors every dimension is a linguistic feature and 1/0 indicates the presence or absence of that feature in a word, thus the vector representations are binary while being highly sparse ($\approx$ 99.9%). Since these vectors do not encode any word cooccurrence information, they are non-distributional. An additional benefit of constructing such vectors is that they are fully interpretable i.e, every dimension of these vectors maps to a linguistic feature unlike distributional word vectors where the vector dimensions have no interpretability.

Of course, engineering feature vectors from linguistic resources is established practice in many applications of discriminative learning; e.g., parsing (McDonald and Pereira, 2006; Nivre, 2008) or part of speech tagging (Ratnaparkhi, 1996; Collins, 2002). However, despite a certain common inventories of features that re-appear across many tasks, feature engineering tends to be seen as a task-specific problem, and engineered feature vectors are not typically evaluated independently of the tasks they are designed for. We evaluate the quality of our linguistic vectors on a number of tasks that have been proposed for evaluating distributional word vectors. We show that linguistic word vectors are comparable to current state-of-the-art distributional word vectors trained on billions of words as evaluated on a battery of semantic and syntactic evaluation benchmarks.[1]

[1]Our vectors can be downloaded at: https://github.com/mfaruqui/non-distributional



*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 464–469,
Beijing, China, July 26-31, 2015. 

| Lexicon | Vocabulary | Features |
|---|---|---|
| WordNet | 10,794 | 92,117 |
| Supersense | 71,836 | 54 |
| FrameNet | 9,462 | 4,221 |
| Emotion | 6,468 | 10 |
| Connotation | 76,134 | 12 |
| Color | 14,182 | 12 |
| Part of Speech | 35,606 | 20 |
| Syn. & Ant. | 35,693 | 75,972 |
| Union | 119,257 | 172,418 |

Table 1: Sizes of vocabualry and features induced from different linguistic resources.

## 2 Linguistic Word Vectors

We construct linguistic word vectors by extracting word level information from linguistic resources. Table 1 shows the size of vocabulary and number of features induced from every lexicon. We now describe various linguistic resources that we use for constructing linguistic word vectors.

**WordNet.** WordNet (Miller, 1995) is an English lexical database that groups words into sets of synonyms called synsets and records a number of relations among these synsets or their members. For a word we look up its synset for all possible part of speech (POS) tags that it can assume. For example, *film* will have SYNSET.FILM.V.01 and SYNSET.FILM.N.01 as features as it can be both a verb and a noun. In addition to synsets, we include the hyponym (for ex. HYPO.COLLAGEFILM.N.01), hypernym (for ex. HYPER:SHEET.N.06) and holonym synset of the word as features. We also collect antonyms and pertainyms of all the words in a synset and include those as features in the linguistic vector.

**Supsersenses.** WordNet partitions nouns and verbs into semantic field categories known as supsersenses (Ciaramita and Altun, 2006; Nastase, 2008). For example, *lioness* evokes the supersense SS.NOUN.ANIMAL. These supersenses were further extended to adjectives (Tsvetkov et al., 2014).[2] We use these supsersense tags for nouns, verbs and adjectives as features in the linguistic word vectors.

**FrameNet.** FrameNet (Baker et al., 1998; Fillmore et al., 2003) is a rich linguistic resource that contains information about lexical and predicate-argument semantics in English. Frames can be realized on the surface by many different word types, which suggests that the word types evoking the same frame should be semantically related. For every word, we use the frame it evokes along with the roles of the evoked frame as its features. Since, information in FrameNet is part of speech (POS) disambiguated, we couple these feature with the corresponding POS tag of the word. For example, since *appreciate* is a verb, it will have the following features: VERB.FRAME.REGARD, VERB.FRAME.ROLE.EVALUEE etc.

**Emotion & Sentiment.** Mohammad and Turney (2013) constructed two different lexicons that associate words to sentiment polarity and to emotions resp. using crowdsourcing. The polarity is either positive or negative but there are eight different kinds of emotions like anger, anticipation, joy etc. Every word in the lexicon is associated with these properties. For example, *cannibal* evokes POL.NEG, EMO.DISGUST and EMO.FEAR. We use these properties as features in linguistic vectors.

**Connotation.** Feng et al. (2013) construct a lexicon that contains information about connotation of words that are seemingly objective but often allude nuanced sentiment. They assign positive, negative and neutral connotations to these words. This lexicon differs from Mohammad and Turney (2013) in that it has a more subtle shade of sentiment and it extends to many more words. For example, *delay* has a negative connotation CON.NOUN.NEG, *floral* has a positive connotation CON.ADJ.POS and *outline* has a neutral connotation CON.VERB.NEUT.

**Color.** Most languages have expressions involving color, for example *green with envy* and *grey with uncertainly* are phrases used in English. The word-color associtation lexicon produced by Mohammad (2011) using crowdsourcing lists the colors that a word evokes in English. We use every color in this lexicon as a feature in the vector. For example, COLOR.RED is a feature evoked by the word *blood*.

**Part of Speech Tags.** The Penn Treebank (Marcus et al., 1993) annotates naturally occurring text for linguistic structure. It contains syntactic parse trees and POS tags for every word in the corpus. We collect all the possible POS tags that a word is annotated with and use it as features in the linguistic vectors. For example, *love* has PTB.NOUN,

[2] http://www.cs.cmu.edu/~ytsvetko/adj-supersenses.tar.gz

| Word | POL.POS | COLOR.PINK | SS.NOUN.FEELING | PTB.VERB | ANTO.FAIR | ··· | CON.NOUN.POS |
|---|---|---|---|---|---|---|---|
| love | 1 | 1 | 1 | 1 | 0 | | 1 |
| hate | 0 | 0 | 1 | 1 | 0 | | 0 |
| ugly | 0 | 0 | 0 | 0 | 1 | | 0 |
| beauty | 1 | 1 | 0 | 0 | 0 | | 1 |
| refundable | 0 | 0 | 0 | 0 | 0 | | 1 |

Table 2: Some linguistic word vectors. 1 indicates presence and 0 indicates absence of a linguistic feature.

PTB.VERB as features.

**Synonymy & Antonymy.** We use Roget's thesaurus (Roget, 1852) to collect sets of synonymous words.[3] For every word, its synonymous word is used as a feature in the linguistic vector. For example, *adoration* and *affair* have a feature SYNO.LOVE, *admissible* has a feature SYNO.ACCEPTABLE. The synonym lexicon contains 25,338 words after removal of multiword phrases. In a similar manner, we also use antonymy relations between words as features in the word vector. The antonymous words for a given word were collected from Ordway (1913).[4] An example would be of *impartiality*, which has features ANTO.FAVORITISM and ANTO.INJUSTICE. The antonym lexicon has 10,355 words. These features are different from those induced from WordNet as the former encode word-word relations whereas the latter encode word-synset relations.

After collecting features from the various linguistic resources described above we obtain linguistic word vectors of length 172,418 dimensions. These vectors are 99.9% sparse i.e, each vector on an average contains only 34 non-zero features out of 172,418 total features. On average a linguistic feature (vector dimension) is active for 15 word types. The linguistic word vectors contain 119,257 unique word types. Table 2 shows linguistic vectors for some of the words.

## 3 Experiments

We first briefly describe the evaluation tasks and then present results.

### 3.1 Evaluation Tasks

**Word Similarity.** We evaluate our word representations on three different benchmarks to measure word similarity. The first one is the widely used WS-353 dataset (Finkelstein et al., 2001), which contains 353 pairs of English words that have been assigned similarity ratings by humans. The second is the RG-65 dataset (Rubenstein and Goodenough, 1965) of 65 words pairs. The third dataset is SimLex (Hill et al., 2014) which has been constructed to overcome the shortcomings of WS-353 and contains 999 pairs of adjectives, nouns and verbs. Word similarity is computed using cosine similarity between two words and Spearman's rank correlation is reported between the rankings produced by vector model against the human rankings.

**Sentiment Analysis.** Socher et al. (2013) created a treebank containing sentences annotated with fine-grained sentiment labels on phrases and sentences from movie review excerpts. The coarse-grained treebank of positive and negative classes has been split into training, development, and test datasets containing 6,920, 872, and 1,821 sentences, respectively. We use average of the word vectors of a given sentence as features in an $\ell_2$-regularized logistic regression for classification. The classifier is tuned on the dev set and accuracy is reported on the test set.

**NP-Bracketing.** Lazaridou et al. (2013) constructed a dataset from the Penn TreeBank (Marcus et al., 1993) of noun phrases (NP) of length three words, where the first can be an adjective or a noun and the other two are nouns. The task is to predict the correct bracketing in the parse tree for a given noun phrase. For example, *local (phone company)* and *(blood pressure) medicine* exhibit *left* and *right* bracketing respectively. We append the word vectors of the three words in the NP in order and use them as features in an $\ell_2$-regularized logistic regression classifier. The dataset contains 2,227 noun phrases split into 10 folds. The classifier is tuned on the first fold and cross-validation accuracy is reported on the remaining nine folds.

[3] `http://www.gutenberg.org/ebooks/10681`

[4] `https://archive.org/details/synonymsantonyms00ordwiala`

| Vector | Length ($D$) | Params. | Corpus Size | WS-353 | RG-65 | SimLex | Senti | NP |
|---|---|---|---|---|---|---|---|---|
| Skip-Gram | 300 | $D \times N$ | 300 billion | 65.6 | 72.8 | 43.6 | **81.5** | 80.1 |
| Glove | 300 | $D \times N$ | 6 billion | 60.5 | 76.6 | 36.9 | 77.7 | 77.9 |
| LSA | 300 | $D \times N$ | 1 billion | **67.3** | 77.0 | 49.6 | 81.1 | 79.7 |
| Ling Sparse | 172,418 | – | – | 44.6 | **77.8** | 56.6 | 79.4 | **83.3** |
| Ling Dense | 300 | $D \times N$ | – | 45.4 | 67.0 | **57.8** | 75.4 | 76.2 |
| Skip-Gram $\oplus$ Ling Sparse | 172,718 | – | – | 67.1 | 80.5 | 55.5 | 82.4 | 82.8 |

Table 3: Performance of different type of word vectors on evaluation tasks reported by Spearman's correlation (first 3 columns) and Accuracy (last 2 columns). Bold shows the best performance for a task.

### 3.2 Linguistic Vs. Distributional Vectors

In order to make our linguistic vectors comparable to publicly available distributional word vectors, we perform singular value decompostion (SVD) on the linguistic matrix to obtain word vectors of lower dimensionality. If $\mathbf{L} \in \{0, 1\}^{N \times D}$ is the linguistic matrix with $N$ word types and $D$ linguistic features, then we can obtain $\mathbf{U} \in \mathbb{R}^{N \times K}$ from the SVD of $\mathbf{L}$ as follows: $\mathbf{L} = \mathbf{U \Sigma V}^{\top}$, with $K$ being the desired length of the lower dimensional space.

We compare both sparse and dense linguistic vectors to three widely used distributional word vector models. The first two are the pre-trained Skip-Gram (Mikolov et al., 2013)[5] and Glove (Pennington et al., 2014)[6] word vectors each of length 300, trained on 300 billion and 6 billion words respectively. We used latent semantic analysis (LSA) to obtain word vectors from the SVD decomposition of a word-word cooccurrence matrix (Turney and Pantel, 2010). These were trained on 1 billion words of Wikipedia with vector length 300 and context window of 5 words.

### 3.3 Results

Table 3 shows the performance of different word vector types on the evaluation tasks. It can be seen that although Skip-Gram, Glove & LSA perform better than linguistic vectors on WS-353, the linguistic vectors outperform them by a huge margin on SimLex. Linguistic vectors also perform better at RG-65. On sentiment analysis, linguistic vectors are competitive with Skip-Gram vectors and on the NP-bracketing task they outperform all distributional vectors with a statistically significant margin ($p < 0.05$, McNemar's test Dietterich (1998)). We append the sparse linguistic vectors to Skip-Gram vectors and evaluate the resultant vectors as shown in the bottom row of Table 3. The combined vector outperforms Skip-Gram on all tasks, showing that linguistic vectors contain useful information orthogonal to distributional information.

It is evident from the results that linguistic vectors are either competitive or better to state-of-the-art distributional vector models. Sparse linguistic word vectors are high dimensional but they are also sparse, which makes them computationally easy to work with.

## 4 Discussion

Linguistic resources like WordNet have found extensive applications in lexical semantics, for example, for word sense disambiguation, word similarity etc. (Resnik, 1995; Agirre et al., 2009). Recently there has been interest in using linguistic resources to enrich word vector representations. In these approaches, relational information among words obtained from WordNet, Freebase etc. is used as a constraint to encourage words with similar properties in lexical ontologies to have similar word vectors (Xu et al., 2014; Yu and Dredze, 2014; Bian et al., 2014; Fried and Duh, 2014; Faruqui et al., 2015a). Distributional representations have also been shown to improve by using experiential data in addition to distributional context (Andrews et al., 2009). We have shown that simple vector concatenation can likewise be used to improve representations (further confirming the established finding that lexical resources and cooccurrence information provide somewhat orthogonal information), but it is certain that more careful combination strategies can be used.

Although distributional word vector dimensions cannot, in general, be identified with linguistic properties, it has been shown that some vector construction strategies yield dimensions that are relatively more interpretable (Murphy et al., 2012; Fyshe et al., 2014; Fyshe et al., 2015; Faruqui et al., 2015b). However, such analysis is difficult to generalize across models of representation. In constrast to distributional word vectors, linguistic

[5] `https://code.google.com/p/word2vec`

[6] `http://www-nlp.stanford.edu/projects/glove/`

word vectors have interpretable dimensions as every dimension is a linguistic property.

Linguistic word vectors require no training as there are no parameters to be optimized, meaning they are computationally economical. While good quality linguistic word vectors may only be obtained for languages with rich linguistic resources, such resources do exist in many languages and should not be disregarded.

## 5 Conclusion

We have presented a novel method of constructing word vector representations solely using linguistic knowledge from pre-existing linguistic resources. These non-distributional, linguistic word vectors are competitive to the current models of distributional word vectors as evaluated on a battery of tasks. Linguistic vectors are fully interpretable as every dimension is a linguistic feature and are highly sparse, so they are computationally easy to work with.

## Acknowledgement

We thank Nathan Schneider for giving comments on an earlier draft of this paper and the anonymous reviewers for their feedback.

## References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proc. of NAACL*.

Mark Andrews, Gabriella Vigliocco, and David Vinson. 2009. Integrating experiential and distributional data to learn semantic representations. *Psychological review*, 116(3):463.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proc. of ACL*.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proc. of ACL*.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Proc. of MLKDD*.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of EMNLP*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015a. Retrofitting word vectors to semantic lexicons. In *Proc. of NAACL*.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015b. Sparse overcomplete word vector representations. In *Proc. of ACL*.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proc. of ACL*.

Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2003. Lexicographic relevance: selecting information from corpus evidence. *International Journal of Lexicography*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proc. of WWW*.

Daniel Fried and Kevin Duh. 2014. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Alona Fyshe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. 2014. Interpretable semantic vectors from a joint model of brain- and text- based meaning. In *Proc. of ACL*.

Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. 2015. A compositional and interpretable semantic space. In *Proc. of NAACL*.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *Proc. of EMNLP*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456.

Angeliki Lazaridou, Eva Maria Vecchi, and Marco Baroni. 2013. Fish transporters and miracle homes: How compositional distributional semantics can help NP parsing. In *Proc. of EMNLP*.

Beth Levin. 1993. *English verb classes and alternations : a preliminary investigation*. University of Chicago Press.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Ryan T McDonald and Fernando CN Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL*.

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

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Saif Mohammad. 2011. Colourful language: Measuring word-colour associations. In *Proc. of the Workshop on Cognitive Modeling and Computational Linguistics*.

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proc. of COLING*.

Vivi Nastase. 2008. Unsupervised all-words word sense disambiguation with grammatical dependencies. In *Proc. of IJCNLP*.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Edith Bertha Ordway. 1913. *Synonyms and Antonyms: An Alphabetical List of Words in Common Use, Grouped with Others of Similar and Opposite Meaning*. Sully and Kleinteich.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of IJCAI*.

P. M. Roget. 1852. *Roget's Thesaurus of English words and phrases*. Available from Project Gutemberg.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10).

Karin Kipper Schuler. 2005. *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

Yulia Tsvetkov, Nathan Schneider, Dirk Hovy, Archna Bhatia, Manaal Faruqui, and Chris Dyer. 2014. Augmenting english adjective senses with supersenses. In *Proc. of LREC*.

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

Peter D. Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proc. of ECML*.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *Proc. of CIKM*.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proc. of ACL*.