# CoKE : Extending Contextualized Knowledge Embeddings for Word Sense Induction

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Word Embeddings are able to capture lexico-semantic information but remain flawed in their inability to assign unique representations to different senses of a polysemous words. They also fail to include information from well curated semantic lexicons and dictionaries. Previous approaches that integrate polysemy and knowledge bases fall distinctly under two categories - a)retrofitting vectors to ontologies or b)learning from sense tagged corpora. While embeddings learned from these methods are superior in understanding contextual similarity, they are outperformed by single prototype word vectors on several relatedness tasks. In this work, we introduce a new approach that can induce polysemy to any pre-trained embedding space by jointly grounding contextualized sense representations and word embeddings to a knowledge base. Along with word sense induction, the resulting representations reduces the effect of vocabulary bias that arises in natural language corpora and in turn embedding spaces. By grounding them to knowledge bases they are able to learn multi-word representations and are also interpretable. We evaluate our vectors across 12 datasets on several similarity and relatedness tasks along with two extrinsic tasks ,we also evaluate against other transfer learning methods and find that our approach consistently outperforms current state of the art.

## 1 Related Work and Introduction

Distributed representations of words (Mikolov et al., 2013b) have proven to be successful in addressing multiple drawbacks of symbolic representations which treats words as atomic units of meaning. By grouping similar words and capturing analogical and lexical relationships, they are a popular choice in several downstream NLP applications.

While these embeddings capture meaningful relationships, they come with their own set of drawbacks. For instance, complete reliance on natural language corpora amplifies existing societal and vocabulary bias that are inherent in datasets. A study by Bolukbasi et al., 2016 discussed societal biases in the form of gender stereotypes present in these VSMs. Vocabulary bias is caused by words not seen in the training corpora and also extends to bias in word usage where some words, often morphologically complex words, are used less frequently than other words or phrases with the same meaning. This also becomes evident in the relatively lower performance of word embeddings on the rare word similarity task (Luong et al., 2013b). A recent approach by (Bojanowski et al., 2016a) proposes using character n-gram representations to address the problem of out-of-vocabulary and rare words. (Faruqui et al., 2014) also proposed retrofitting vectors to an ontology. However, these methods don't account for polysemy.

Polysemy is an important feature of language which causes words to have different meanings based on the context in which they occur. For instance, the word 'bank' can mean 'financial institution' or

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

'land on either side of a river'.A well known drawback with word embeddings is the assignment of a single vector representation to a word type, irrespective of polysemy. A large body of work has gone into developing word sense disambiguation systems to identify the correct sense of a word based on it's context. The availability of such disambiguation systems coupled with the growing reliance of NLP systems on distributional semantics has led to an increasing interest in obtaining powerful sense representations.

Some of the previous work that has gone into learning sense representations includes unsupervised learning techniques to cluster contexts and learn multi prototype vectors(Reisinger and Mooney (2010) , Huang et al. (2012) and Wu and Giles (2015)). However, a common drawback with the cluster based models is the difficulty in deciding the number of clusters apriori. ( Neelakantan et al. (2015) , Tian et al. (2014) ,Cheng and Kartsaklis (2015) also learn multiple word embeddings by modifying the Skip-Gram model. These approaches yield to sense representations that are limited in terms of interpretability which makes it challenging to include in downstream tasks.

As a remedy to limitation in sense tagged corpora, Jauhar et al. (2015) and Rothe and Schütze (2015) explored grounding word embeddings to ontologies to obtain sense representations. As a result, these techniques drastically improved performance on several similarity tasks but an observed pattern is that this leads to compromised performance on word relatedness tasks(Faruqui et al. (2014), Jauhar et al. (2015)). We suspect this is a result of directly modifying word embedding spaces based on ontology structure.

In this work, we present a novel approach that uses knowledge bases and sense representations to directly induce polysemy to any predefined word embedding space. We show how our approach allows the integration of ontological information and leads to improvements in both word similarity and relatedness tasks. The advantages of this are plenty, it allows the integration of knowledge into embedding spaces and can readily induce polysemy in them. We thus rely on a) Sense tagged corpora to obtain contextualized sense representations. The objective of which is to capture sense relations and interactions in naturally occurring corpora. The sense representations are interpretable and have lexical mappings to a knowledge base. We use them to induce polysemy in word embedding spaces. b) Pretrained word embeddings to capture many useful lexical relationships that are inherent in them on account of being trained on large amounts of data. These relationships are not effectively captured by sense representations due to the limited size of sense tagged corpora they are trained on. c) Lastly, in order to account for the vocabulary bias which causes similar meaning words to be farther apart in embedding spaces as a result of bias in co-occurrence statistics found in corpora, we use a knowledge base to jointly ground word and sense representations.
We thus obtain unique multiple word sense representations that show superior performance in similarity, relatedness and extrinsic tasks. They also show performance benefits when used with transfer learning methods like CoVE (McCann et al. (2017)) and ELMo (Peters et al. (2018))

## 2 Methodology

### 2.1 Lexicon Building

We use WordNet(Miller (1995)) as our primary knowledge base source. However, in order to obtain representations that cater to both similarity and relatedness, we modify the synset nodes in WordNet. A synset in WordNet is represented by a set of synonyms. We observe that these synonym sets include words of same meaning without differentiating between their syntactic forms. For instance, the synset *operate.v.01*, defined as 'direct or control; projects, businesses' has both *run* and *running* in its synonym sets. In practice though, each syntactic form of a word has different semantic distributions. For instance, in this case *run* is found to most likely occur with words such as *lead* and *head* as compared to it's alternate form *running* which is more likely to appear with words such as *managing*, *administrating* , *leading*. To account for this, we extend WordNet nodes to also include the syntactic form information and call a synset, syntactic form pair "sense-forms". The extended WordNet nodes is depicted in Figure 1.

### 2.2 Sense-Form Embeddings

We use a concatenation of two sense tagged corpora, SemCor(Ciaramita and Altun (2006)) and OMSTI(Taghipour and Ng (2015)) to obtain sense representations. To better capture different

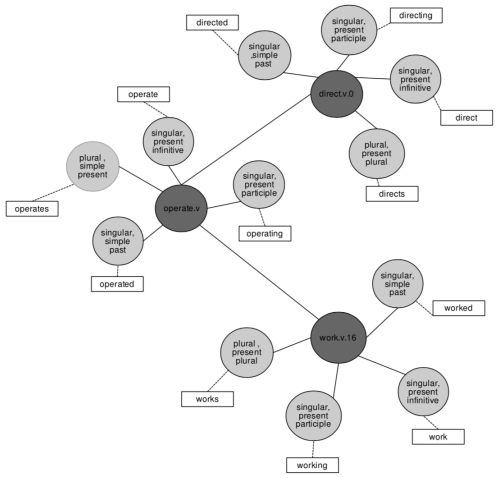


Figure 1: Example WordNet nodes extended to include sense forms

relational and interaction information between senses, we pre-process the corpora by replacing every word and synset pair with it's respective sense tag and syntactic form. We then use the Word2Vec toolkit(Mikolov et al. (2013b)) with the Skip Gram objective function with Negative Sampling to obtain our "sense-form" representations.

### 2.3 Thesaurus Inclusion

While WordNet provides valuable structural information, the stringent structure leads to a rather limited synonym set. We thus make use of an external thesaurus [1]to augment each synonym set.

### 2.4 Word Sense Induction

Now that we have sense and word representations, we ground the two in WordNet and obtain multi word sense representations. Thus, for given a word with word embedding $v_w$ and sense-form with sense-form embedding $v_s$, we obtain unique word specific sense representations $v_{s_w}$ as follows.

$$v_{s_w} = P(senseform|word)v_s$$

Arora et al. (2018) infact observe that a word vector lies in the linear superposition of it's senses as :

$$v_{word} = \alpha_1 v_{word\_sense1} + \alpha_2 v_{word\_sense2} + \alpha_3 v_{word\_sense3} + ...$$

The coefficients contributing to each sense of a word is the ratio of frequency of the word senses. Thus given a word with 2 senses, the coefficients follow a probability distribution as $1 - clog(r)$ where r is the ratio of frequencies of the two senses for some constant c, where $r < 10^{1/c}$. We thus mimic this property and ground our sense representations to WordNet to obtain :

$$P(senseform|word) = 1 - clog(rank(senseform|word))$$

To start of, we first rank all the sense-forms a given word is found in decreasing order of likelihood. WordNet ranks synsets for a word based on the frequency that a word has occurred with respect to a synset sense. We thus exploit this property of WordNet to rank sense forms. Since we use an external thesaurus to augment synonym sets, a word typically has senseform not found in the original WordNet list. To account for this, we use the ontology structure to find the closest synset from the original list

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

to the senseform and use it's rank. By conditioning rank on structure, this leads to ontology grounded sense forms. Once we obtain these word specific sense representations, we transfer this to pre existing embedding spaces to induce polysemy. To best combine information from the two vectors spaces, we experiment with two vector manipulations, concatenation and modification.
1) Concatenation : We simply concatenate the word specific senseform vectors to the respective word's pretrained word embedding

$$v_{w,s} = [v_w; v_{s_w}]$$

2) Modification :

$$v_{w,s} = [v_w; v_{s_w}; v_w - v_{s_w}; v_w \odot v_{s_w}]$$

The combination is thus done as a post processing step after obtaining word specific sense representations.

# 3 Experimental Setup

In this section, we describe the experiments done to evaluate our multi sense word embeddings. We use an array of existing word similarity and relatedness datasets to conduct intrinsic evaluation and 4 datasets across 2 tasks for extrinsic evaluation.

## 3.1 Intrinsic Evaluation

### 3.1.1 Word Representations

We pick three different embeddings(Pennington et al. (2014) , Bojanowski et al. (2016b) , Mikolov et al. (2013a)) of 300 dimension to run our experiments

### 3.1.2 Similarity Measures

Given a pair of words $w$ with $M$ senses and $w^{'}$ with $N$ senses, we follow Reisinger and Mooney (2010) and use two metrics for computing similarity scores without using context.

$$AvgSim(w, w^{'}) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (cos(v_{w,i}, v_{w^{'},j}))$$

$$MaxSim(w, w^{'}) = \max_{1 \leq i \leq M, 1 \leq j \leq M} cos(v_{w,i}, v_{w^{'},j})$$

$AvgSim$ computes word similarity as the average similarity between all pairs of sense vectors. Whereas $MaxSim$ computes the maximum similarity over all pairwise sense vector similarities.

### 3.1.3 Word Similarity

We evaluate our embeddings on several standard word similarity datasets namely, SimLex (Hill et al. (2015)), WordSim-353(Gabrilovich and Markovitch), WS-S, MC-30(Miller and Charles (1991)) , RG-65 (Rubenstein and Goodenough (1965)), YP-130 (Yang and Powers (2006)),SimVerb(Gerz et al. (2016))and Rare Word(RW) similarity (Luong et al. (2013a)).
Each dataset contains a list of word pairs with a human score of how related or similar the two words are.We calculate the Spearman correlation between the labels and the scores generated by our method.We evaluate with another baseline using random vectors as senseform vectors along with word embeddings. $MaxSim$ shows bigger improvements on all datasets, except for WordSim -353 and Rare Word and we thus use $AvgSim$ for it. The results are outlined in Table 1.

### 3.1.4 Word Relatedness

Integration of our vectors also shows improvements in word relatedness tasks.As our benchmark, we evaluate on WS-R (relatedness) , MTurk(771) (Halawi et al. (2012)), MEN(Bruni et al.

| Vector | WS-S | RG-65 | RW | SL-999 | YP | MC | SV-3500 |
|---|---|---|---|---|---|---|---|
| **Skip-Gram** | **76.23** | 76.60 | 47.13 | 45.39 | 58.02 | 78.34 | 37.54 |
| +RANDOM | 52.6 | 60.77 | 28.92 | 33.59 | 44.48 | 70.76 | 25.20 |
| +CoKE(C) | 75.33 | 84.41 | 48.82 | **62.13** | **67.86** | **80.68** | **51.50** |
| +CoKE(M) | 73.37 | **84.65** | **48.91** | 62.10 | 67.62 | 80.47 | 51.42 |
| **Glove** | 79.86 | 76.60 | 43.72 | 43.66 | 55.51 | 78.87 | 29.38 |
| +RANDOM | 52.58 | 60.75 | 28.5 | 32.63 | 44.58 | 69.27 | 18.60 |
| +CoKE(C) | 81.19 | **84.53** | 45.81 | **57.25** | **63.67** | 82.7 | **42.93** |
| +CoKE(M) | **81.28** | 84.03 | **45.9** | 56.78 | **64.18** | **82.51** | 42.33 |
| **FastText** | **77.42** | 79.25 | 45.60 | 38.51 | 56.31 | 83.14 | 26.32 |
| +RANDOM | 60.07 | 76.02 | 33.35 | 31.1 | 46.6 | 64.11 | 14.30 |
| +CoKE(C) | **76.09** | **87.14** | **47.02** | **59.32** | **66.71** | **85.56** | **47.44** |
| +CoKE(M) | 76.14 | 86.73 | 46.88 | 59.18 | 66.55 | 85.23 | 47.41 |

Table 1: Spearman Correlation on Word Similarity Task..(Higher scores are better)

| Vector | WS-R | MEN | MT-771 | SGS |
|---|---|---|---|---|
| SG | 58.11 | **72.45** | **65.21** | 57.15 |
| +RANDOM | 44.58 | 59.94 | 30.46 | 40.02 |
| +CoKE(C) | **58.46** | 71.94 | 62.52 | 60.82 |
| +CoKE(M) | 58.32 | 71.92 | 62.66 | **61.39** |
| Glove | 68.31 | 79.80 | 70.51 | 60.09 |
| +RANDOM | 45.14 | 64.48 | 53.07 | 30.1 |
| +CoKE(C) | **69.48** | **79.86** | 71.93 | **71.46** |
| +CoKE(M) | 69.35 | 79.53 | **72.00** | 71.23 |
| FastText | **65.49** | **75.3** | 65.19 | 55.40 |
| +RANDOM | 53.46 | 62.80 | 52.10 | 40.02 |
| +CoKE(C) | 65.37 | 74.73 | **65.52** | 64.67 |
| +CoKE(M) | 65.18 | 75.1 | 63.24 | **64.75** |

Table 2: Spearman Correlation on Word Relatedness Task

| **Model** | $\rho$ **x 100** |
|---|---|
| Jauhar et al. (2015) | 61.3 |
| Iacobacci et al. (2015) , 2015 | 62.4 |
| Huang et al. (2012) | 62.8 |
| Athiwaratkun and Wilson (2017) | 65.5 |
| *CoKE + SG(Our model)* | *65.9* |
| Chen et al. (2014) | 66.2 |
| Rothe and Schutze (2015) | 68.9 |

Table 3: Spearman Correlation on Stanford Contextual Word Similarity Dataset.(Higher scores are better)

(2012)),SGS130 ( Szumlanski et al. (2013)), this dataset also includes phrases. We evaluate the performance of our method against standard pretrained word embedding using spearman correlation. We depict the performance of $MaxSim$ on MT(771) and SGS130 and $AvgSim$ for WS-R and MEN. The results are outlined in Table 2.

#### 3.1.5 Word Similarity for Polysemous Words

We use the SCWS dataset introduced by Luong et al. (2013a), where word pairs are chosen to have variations in meanings for polysemous and homonymous words. We compare our method with other state of the art multiprototype models .We find that our model performs competitively with previous models. We use the SkipGram(SG) word embedding with our method to allow for fair comparison

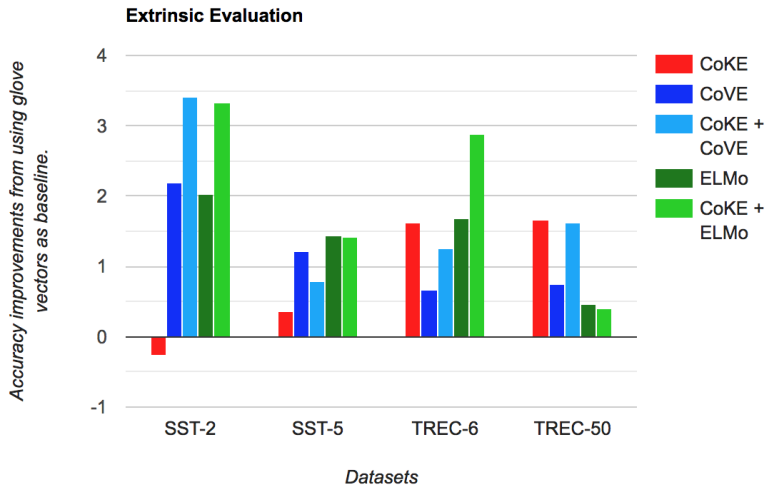


Figure 2: Performance improvements with CoKE.

with the previous methods listed which uses SkipGram based retrofitting to WordNet.The spearman correlation between the labels and scores are seen in Table 3.

## 3.2 Extrinsic Evaluation

### 3.2.1 Datasets

For sentiment analysis we use the Stanford Sentiment Treebank dataset(Socher et al. (2013)). We train seperately and test on the Binary Version(SST-2) as well as the five class version(SST-5). For question classification, we evaluate performance on the TREC(Voorhees (2001)) question classification dataset which consists of open domain questions and semantic categories.

| Dataset | GloVe | CoKE | CoVE | CoKE(+CoVE ) | ELMo | CoKE(+ELMo) |
|---|---|---|---|---|---|---|
| **SST-2** | 85.99 | 85.72 | 88.18 | **89.41** | 88.02 | 89.32 |
| **SST-5** | 50.19 | 50.56 | 51.4 | 50.97 | **51.62** | 51.60 |
| **TREC-6** | 89.90 | 91.53 | 90.56 | 91.15 | 91.59 | **92.78** |
| **TREC-50** | 83.84 | **85.5** | 84.59 | 85.46 | 84.31 | 84.249 |

Table 4: CoKE improves performance when used alone as well as when used with a disambiguation system. Note, CoVE and ELMo are only used for disambiguation, their representations aren't included with CoKE(Higher scores are better)

### 3.2.2 Performance Comparisons

We conduct two experiments to evaluate the usefulness of our vectors. The first comparison is against using pre-trained word embeddings alone(Pennington et al. (2014)). For this experiment, we represent each word as the average of all it's sense vectors learned by CoKE.

Recent trends have also lead to an increasing interest in transfer learning. Both CoVE( McCann et al. (2017)) and ELMo( Peters et al. (2018) )show significant improvements in extrinsic tasks by inclusion with pre-trained word embeddings. As shown by Peters et al. (2018), these systems inherently act as word sense disambiguation and representation systems. They give word representations conditioned on the context it occurs in. However, they rely entirely on distributional semantics and could benefit from including knowledge base information. Thus in our second experiment, we first train and test using vanilla CoVE and ELMo representations. We then use them as disambiguation systems but

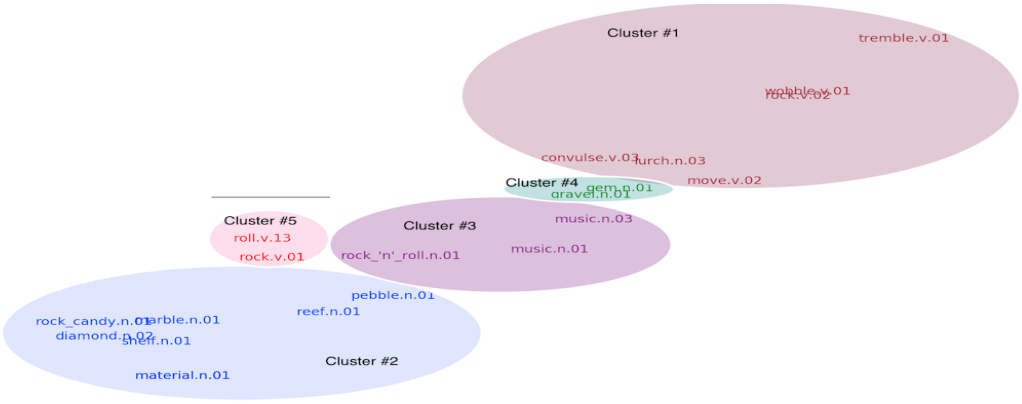


Figure 3: Sense clusters for "rock"

replace word representations with CoKE embeddings instead and show performance improvements. For disambiguating using CoKE and ELMo, we use the same approach as outlined in Peters et al. (2018). We first use the sense tagged corpus to compute CoVE or ELMo word representations and then take the average representation for each sense to obtain sense representations. During disambiguation, we again use CoVE or ELMo architecture to compute representations for a given word and then take the nearest neighbor sense from the training set to get the correct sense. We then use respective CoKE embeddings on the disambiguated data.

#### 3.2.3 Training Details

To test for performance of different embeddings on datasets, we implement a an LSTM(Hochreiter and Schmidhuber (1997)) with a hidden size of 300 and run our experiments. Parameters were fine-tuned specific to task and embedding type.

## 4 Qualitative Analysis

In this section, we look at some visualizations of senses induced and show how they are easily interpretable. Since the sense tags have lexical mappings to an ontology,they can be looked up to find meanings. Moreover, the semantic distribution also plays a role in obtaining meaningful sense clusters. We analyze two things 1) Sense clusters induced 2)How using different sense forms affect representations and sense interactions in their respective word forms. For all our analysis, we use the concatenated version of CoKE + GLoVE embeddings and do a dimensionality reduction.

### 4.1 Sense Clusters

: We look at the sense clusters formed by our word specific senses embeddings for the word "rock".

The clusters for the word "rock" is depicted in Fig2, the multiple fine grained embeddings cluster to form 5 basic sense meanings. We see three distinct clusters that dominate. "Cluster#2" can be interpreted as all synsets that speak of rock as a "substance". In , "Cluster#3", the synsets cluster together to speak of rock as "music". An interesting property can be observed comparing "Cluster#1" and "Cluster#5" . The senses found in both of these clusters interpret "rock" as "movement/motion". However the two distinct clusters also capture the kind of motion . For instance ,the senses $roll.v.13$ and $rock.v.01$ in "Cluster#5" map specifically "sideways movement". While the senses in "Cluster#1" map to glosses "sudden movements"(convulse , lurch,move , tremble) and "back and forth movements(wobble , rock)". Another interesting property is depicted by "Cluster#4", although they are more synonymous in meaning to rock as a "substance", the senses for gravel cluster very closely to senses mapping to gloss "jerking" movements capturing deeper relations between sense.

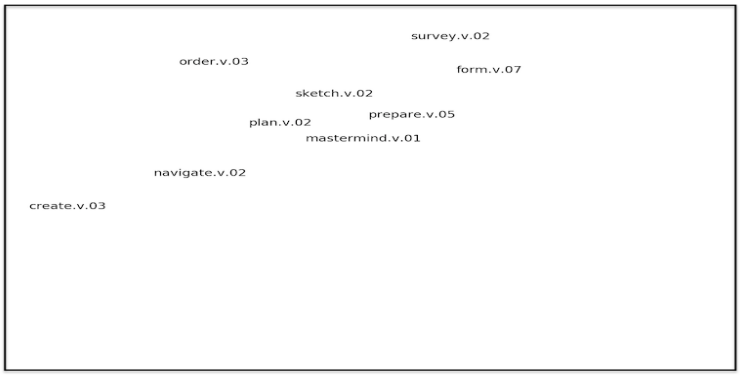


Figure 4: Sense interaction of mastermind.v.01 for the word "plan"


Figure 5: Sense interaction of mastermind.v.01 for the word "planning"

### 4.2 Sense Forms

In this section, we analyze how different sense form representations interact for synonyms within a sense. We do so by considering the word forms "plan" and "planning" both of which are synonyms of their respective senseforms of "mastermind.v.01" ( Gloss : plan and direct ,a complex undertaking).

In order to observe difference in relationships of word forms, we consider only common synsets in "plan" and "planning" for visualization and observe the interactions with each other. For the word "plan" as shown in Figure 5, we observe that the synset "mastermind" is closer in proximity to synsets that map to words like "plan" , "sketch" , "prepare". In contrast the same synset in the embedding space for "planning" as shown in Figure 6 interacts closely with synsets that are analogous to "project planning" , "scheduling" , "organizing". This shows how using different sense form representations, leads to different interactions among the same group of synsets for different words and allows for better interaction that is more unique to each word.

## 5 Conclusion

In our work, we explore the possibility of obtaining multiword prototypes from embedding spaces by directly transfer learning from an ontology. The prototypes allow ease of use with WSD systems , can easily be used in downstream applications since they are portable and are flexible to use in a wide variety of tasks. While previous work on polysemy falls under three distinct clusters of learning multi word sense representations, resource specific sense vectors and grounding vectors directly to lexicons. Our work lies in the intersection.

# References


Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association of Computational Linguistics*, 6:483–495.

Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1645–1656.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016a. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016b. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *arXiv preprint arXiv:1508.02354*.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 594–602. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint arXiv:1608.00869*.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 95–105.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 683–693.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013a. Better word representations with recursive neural networks for morphology. In *CoNLL*.

Thang Luong, Richard Socher, and Christopher Manning. 2013b. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

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 S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *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.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

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*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

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. Association for Computational Linguistics.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Sean Szumlanski, Fernando Gomez, and Valerie K Sims. 2013. A new set of norms for semantic relatedness measures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 890–895.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160.

Ellen M Voorhees. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.

Zhaohui Wu and C Lee Giles. 2015. Sense-aaware semantic analysis: A multi-prototype word representation model using wikipedia. In *AAAI*, pages 2188–2194.

Dongqiang Yang and David Martin Powers. 2006. *Verb similarity on the taxonomy of WordNet*. Masaryk University.