Sense Embedding Learning for Word Sense Induction

Linfeng Song[1], **Zhiguo Wang**[2], **Haitao Mi**[2] and **Daniel Gildea**[1]
[1]Department of Computer Science, University of Rochester, Rochester, NY 14627
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

Abstract

Conventional word sense induction (WSI) methods usually represent each instance with discrete linguistic features or co-occurrence features, and train a model for each polysemous word individually. In this work, we propose to learn sense embeddings for the WSI task. In the training stage, our method induces several sense centroids (embedding) for each polysemous word. In the testing stage, our method represents each instance as a contextual vector, and induces its sense by finding the nearest sense centroid in the embedding space. The advantages of our method are (1) distributed sense vectors are taken as the knowledge representations which are trained discriminatively, and usually have better performance than traditional count-based distributional models, and (2) a general model for the whole vocabulary is jointly trained to induce sense centroids under the mutli-task learning framework. Evaluated on SemEval-2010 WSI dataset, our method outperforms all participants and most of the recent state-of-the-art methods. We further verify the two advantages by comparing with carefully designed baselines.

1 Introduction

Word sense induction (WSI) is the task of automatically finding sense clusters for polysemous words. In contrast, word sense disambiguation (WSD) assumes there exists an already-known sense inventory, and the sense of a word type is disambiguated according to the sense inventory. Therefore, clustering methods are generally applied in WSI tasks, while classification methods are utilized in WSD tasks. WSI has been successfully applied to many NLP tasks such as machine translation (Xiong and Zhang, 2014), information retrieval (Navigli and Crisafulli, 2010) and novel sense detection (Lau et al., 2012).

However, existing methods usually represent each instance with discrete hand-crafted features (Bordag, 2006; Chen et al., 2009; Van de Cruys and Apidianaki, 2011; Purandare and Pedersen, 2004), which are designed manually and require linguistic knowledge. Most previous methods require learning a specific model for each polysemous word, which limits their usability for down-stream applications and loses the chance to jointly learn senses for multiple words.

There is a great advance in recent distributed semantics, such as word embedding (Mikolov et al., 2013; Pennington et al., 2014) and sense embedding (Reisinger and Mooney, 2010; Huang et al., 2012; Jauhar et al., 2015; Rothe and Schütze, 2015; Chen et al., 2014; Tian et al., 2014). Comparing with word embedding, sense embedding methods learn distributed representations for senses of a polysemous word, which is similar to the sense centroid of WSI tasks.

In this work, we point out that the WSI task and the sense embedding task are highly inter-related, and propose to jointly learn sense centroids (embeddings) of all polysemous words for the WSI task. Concretely, our method induces several sense centroids (embedding) for each polysemous word in training stage. In testing stage, our method represents each instance as a contextual vector, and induces its sense by finding the nearest sense centroid in the embedding space. Comparing with existing methods, our method has two advantages: (1) distributed sense embeddings are taken as the knowledge representations which are trained discriminatively, and usually have better performance than traditional count-based dis-

*Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM 2016)*, pages 85–90,
Berlin, Germany, August 11-12, 2016.

tributional models (Baroni et al., 2014), and (2) a general model for the whole vocabulary is jointly trained to induce sense centroids under the mutli-task learning framework (Caruana, 1997). Evaluated on SemEval-2010 WSI dataset, our method outperforms all participants and most of the recent state-of-the-art methods.

2 Methodology

2.1 Word Sense Induction

WSI is generally considered as an unsupervised clustering task under the distributional hypothesis (Harris, 1954) that the word meaning is reflected by the set of contexts in which it appears. Existing WSI methods can be roughly divided into feature-based or Bayesian. Feature-based methods first represent each instance as a context vector, then utilize a clustering algorithm on the context vectors to induce all the senses. Bayesian methods (Brody and Lapata, 2009; Yao and Van Durme, 2011; Lau et al., 2012; Goyal and Hovy, 2014; Wang et al., 2015), on the other hand, discover senses based on topic models. They adopt either the LDA (Blei et al., 2003) or HDP (Teh et al., 2006) model by viewing each target word as a corpus and the contexts as pseudo-documents, where a context includes all words within a window centred by the target word. For sense induction, they first extract pseudo-documents for the target word, then train topic model, finally pick the most probable topic for each test pseudo-document as the sense.

All of the existing WSI methods have two important factors: 1) how to group similar instances (clustering algorithm) and 2) how to represent context (knowledge representation). For clustering algorithms, feature-based methods use k-means or graph-based clustering algorithms to assign each instance to its nearest sense, whereas Bayesian methods sample the sense from the probability distribution among all the senses for each instance, which can be seen as soft clustering algorithms. As for knowledge representation, existing WSI methods use the vector space model (VSM) to represent each context. In feature-based models, each instance is represented as a vector of values, where a value can be the count of a feature or the co-occurrence between two words. In Bayesian methods, the vectors are represented as co-occurrences between documents and senses or between senses and words. Overall existing methods separately train a specific VSM for each word. No methods have shown distributional vectors can keep knowledge for multiple words while showing competitive performance.

2.2 Sense Embedding for WSI

As mentioned in Section 1, sense embedding methods learn a distributed representation for each sense of a polysemous word. There are two key factors for sense embedding learning: (1) how to decide the number of senses for each polysemous word and (2) how to learn an embedding representation for each sense. To decide the number of senses in factor (1), one group of methods (Huang et al., 2012; Neelakantan et al., 2014) set a fixed number K of senses for each word, and each instance is assigned to the most probable sense according to Equation 1, where $\mu(w_t, k)$ is the vector for the k-th sense centroid of word w, and v_c is the representation vector of the instance.

$$s_t = \arg\max_{k=1,..,K} sim(\mu(w_t, k), v_c) \qquad (1)$$

Another group of methods (Li and Jurafsky, 2015) employs non-parametric algorithms to dynamically decide the number of senses for each word, and each instance is assigned to a sense following a probability distribution in Equation 2, where S_t is the set of already generated senses for w_t, and γ is a constant probability for generating a new sense for w_t.

$$s_t \sim \begin{cases} p(k|\mu(w_t, k), v_c) \; \forall \; k \in S_t \\ \gamma \;\; \text{for new sense} \end{cases} \qquad (2)$$

From the above discussions, we can obviously notice that WSI task and sense embedding task are inter-related. The two factors in sense embedding learning can be aligned to the two factors of WSI task. Concretely, deciding the number of senses is the same problem as the clustering problem in WSI task, and sense embedding is a potential knowledge representation for WSI task. Therefore, sense embedding methods are naturally applicable to WSI.

In this work, we apply the sense embedding learning methods for WSI tasks. Algorithm 1 lists the flow of our method. The algorithm iterates several times over a Corpus (Line 2-3). For each token w_t, it calculates the context vector v_c (Line 4) for an instance, and then gets the most possible

Algorithm 1 Sense Embedding Learning for WSI

```
1: procedure TRAINING(Corpus $C$)
2:     for $iter$ in [1..$I$] do
3:         for $w_t$ in $C$ do
4:             $v_c \leftarrow$ context_vec($w_t$)
5:             $s_t \leftarrow$ sense_label($w_t$, $v_c$)
6:             update($w_t$, $s_t$)
7:         end for
8:     end for
9: end procedure
```

sense label s_t for w_t (Line 5). Finally, both the sense embeddings for s_t and global word embeddings for all context words of w_t are updated (Line 6). We introduce our strategy for *context_vec* in the next section. For *sense_label* function, a sense label is obtained by either Equation 1 or Equation 2. For the *update* function, vectors are updated by the Skip-gram method (same as Neelakantan et al. (2014)) which tries to predict context words with the current sense. In this algorithm, the senses of all polysemous words are learned jointly on the whole corpus, instead of training a single model for each individual word as in the traditional WSI methods. This is actually an instance of multi-task learning, where WSI models for each target word are trained together, and all of these models share the same global word embeddings.

Comparing to the traditional methods for WSI tasks, the advantages of our method include: 1) WSI models for all the polysemous words are trained jointly under the multi-task learning framework; 2) distributed sense embeddings are taken as the knowledge representations which are trained discriminatively, and usually have better performance than traditional count-based distributional models (Baroni et al., 2014). To verify the two statements, we carefully designed comparative experiments described in the next section.

3 Experiment

3.1 Experimental Setup and baselines

We evaluate our methods on the test set of the SemEval-2010 WSI task (Manandhar et al., 2010). It contains 8,915 instances for 100 target words (50 nouns and 50 verbs) which mostly come from news domain. We choose the April 2010 snapshot of Wikipedia (Shaoul and Westbury, 2010) as our training set, as it is freely available and domain general. It contains around 2 million documents and 990 million tokens. We train and test our models and the baselines according to the above data setting, and compare with reported performance on the same test set from previous papers.

For our sense embedding method, we build two systems: *SE-WSI-fix* which adopts Multi-Sense Skip-gram (MSSG) model (Neelakantan et al., 2014) and assigns 3 senses for each word type, and *SE-WSI-CRP* (Li and Jurafsky, 2015) which dynamically decides the number of senses using a Chinese restaurant process. For *SE-WSI-fix*, we learn sense embeddings for the top 6K frequent words in the training set. For *SE-WSI-CRP*, we first learn word embeddings with word2vec[1], then use them as pre-trained vectors to learn sense embeddings. All training is under default parameter settings, and all word and sense embeddings are fixed at 300 dimensions. For fair comparison, we create *SE-WSI-fix-cmp* by training the MSSG model on the training data of the SemEval-2010 WSI task with the same setting of *SE-WSI-fix*.

We also design baselines to verify the two advantages of our sense embedding methods. One (*CRP-PPMI*) uses the same CRP algorithm as *SE-WSI-CRP*, but with Positive PMI vectors as pre-trained vectors. The other (*WE-Kmeans*) uses the vectors learned by *SE-WSI-fix*, but separately clusters all the context vectors into 3 groups for each target word with kmeans. We compute a context vector by averaging the vectors of all selected words in the context[2].

3.2 Comparing on SemEval-2010

We compare our methods with the following systems: (1) *UoY* (Korkontzelos and Manandhar, 2010) which is the best system in the SemEval-2010 WSI competition; (2) *NMF*$_{lib}$ (Van de Cruys and Apidianaki, 2011) which adopts non-negative matrix factorization to factor a matrix and then conducts word sense clustering on the test set; (3) *NB* (Choe and Charniak, 2013) which adopts naive Bayes with the generative story that a context is generated by picking a sense and then all context words given the sense; and (4) *Spectral* (Goyal and Hovy, 2014) which applies spectral clustering on a set of distributional context vectors.

Experimental results are shown in Table 1. Let us see the results on supervised recall (80-20 SR)

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

[2] A word is selected only if its length is greater than 3, not the target word, or not in a self-constructed stoplist.

System	V-Measure(%)			Paired F-score(%)			80-20 SR(%)			FS	#CI
	All	Noun	Verb	All	Noun	Verb	All	Noun	Verb	All	
UoY (2010)	15.7	20.6	8.5	49.8	38.2	66.6	62.4	59.4	66.8	-	11.5
NMF$_{lib}$ (2011)	11.8	13.5	9.4	45.3	42.2	49.8	62.6	57.3	70.2	-	4.80
NB (2013)	**18.0**	**23.7**	**9.9**	52.9	52.5	53.5	65.4	62.6	69.5	-	3.42
Spectral (2014)	4.5	4.6	4.2	**61.5**	**54.5**	**71.6**	-	-	-	60.7	1.87
SE-WSI-fix-cmp	16.3	20.8	9.7	54.3	54.2	54.3	**66.3**	**63.6**	**70.2**	**66.4**	2.61
SE-WSI-fix	9.8	13.5	4.3	55.1	50.7	61.6	62.9	58.5	69.2	63.0	2.50
SE-WSI-CRP	5.7	7.4	3.2	55.3	49.4	63.8	61.2	56.3	67.9	61.3	2.09
CRP-PPMI	2.9	3.5	2.0	57.7	53.3	64.0	59.2	53.6	67.4	59.2	1.76
WE-Kmeans	4.6	5.0	4.1	51.2	46.5	57.6	58.6	53.3	66.4	58.6	2.54

Table 1: Result on SemEval-2010 WSI task. *80-20 SR* is the supervised recall of 80-20 split supervised evaluation. *FS* is the F-Score of 80-20 split supervised evaluation. *#CI* is the average number of clusters (senses)

first, as it is the main indicator for the task. Overall, *SE-WSI-fix-cmp*, which jointly learns sense embedding for 6K words, outperforms every comparing systems which learns for each single word. This shows that sense embedding is suitable and promising for the task of word sense induction. Trained on out-of-domain data, *SE-WSI-fix* outperforms most of the systems, including the best system in the shared task (*UoY*), and *SE-WSI-CRP* works better than *Spectral* and all the baselines. This also shows the effectiveness of the sense embedding methods. Besides, *SE-WSI-CRP* is 1.7 points lower than *SE-WSI-fix*. We think the reason is that *SE-WSI-CRP* induces fewer senses than *SE-WSI-fix* (see the last column of Table 1). Since both systems induce fewer senses than the golden standard which is 3.85, inducing fewer senses harms the performance. Finally, simple as it is, *NB* shows a very good performance. However *NB* can not benefit from large-scale data as its number of parameters is small, and it uses EM algorithm which is generally slow. Sense embedding methods have other advantages that they train a general model while *NB* learns specific model for each target word.

As for the unsupervised evaluations, *SE-WSI-fix* achieves a good V-Measure score (VM) with a few induced senses. Pedersen (2010) points out that bad models can increase VM by increasing the number of clusters, but doing this will harm performance on both Paired F-score (PF) and SR. Even though *UoY*, *NMF$_{lib}$* and *NB* show better VM, they (especially *UoY*) induced more senses than *SE-WSI-fix*. *SE-WSI-fix* has higher PF than all others, and higher SR than *UoY* and *NMF$_{lib}$*. Trained on the official training data of SemEval-2010 WSI task, *SE-WSI-fix-cmp* achieves the top performance on both VM and PF, while it induces a reasonable number of averaged senses. Comparatively *SE-WSI-CRP* has lower VM and induces fewer senses than *SE-WSI-fix*. One possible reason is that the “rich gets richer” nature of CRP makes it conservative for making new senses. But its PF and SR show that it is still a highly competitive system.

To verify the advantages of our method, we first compare *SE-WSI-CRP* with *CRP-PPMI* as their only difference is the vectors for representing contexts. We can see that *SE-WSI-CRP* performs significantly better than *CRP-PPMI* on both SR and VM. *CRP-PPMI* has higher PF mainly because it induces fewer number of senses. The above results prove that using sense embeddings have better performance than using count-based distributional models. Besides, *SE-WSI-fix* is significantly better than *WE-Kmeans* on every metric. As *WE-Kmeans* and *SE-WSI-fix* learn sense centroids in the same vectors space, while the latter performs joint learning. Therefore, the joint learning is better than learning separately.

4 Related Work

Kågebäck et al. (2015) proposed two methods to utilize distributed representations for the WSI task. The first method learned centroid vectors by clustering all pre-computed context vectors of each target word. The other method simply adopted *MSSG* (Neelakantan et al., 2014) and changed context vector calculation from the average of all context word vectors to weighted average. Our work has further contributions. First, we clearly point out the two advantages of sense embedding methods: 1) joint learning under the mutli-task learning framework, 2) better knowledge representation by discriminative training, and verify them by experiments. In addition, we adopt various sense embedding methods to show that sense embedding methods are generally promising for WSI, not just one method is better than other methods. Finally, we compare our methods with recent state-of-the-art WSI methods on both supervised and unsupervised metrics.

5 Conclusion

In this paper, we show that sense embedding is a promising approach for WSI by adopting two different sense embedding based systems on the SemEval-2010 WSI task. Both systems show highly competitive performance while they learn a general model for thousands of words (not just the tested polysemous words). we believe that the two advantages of our method are: 1) joint learning under the mutli-task learning framework, 2) better knowledge representation by discriminative training, and verify them by experiments.

Acknowledgments

Funded by NSF IIS-1446996. We would like to thank Yue Zhang for his insightful comments on the first version of the paper, and the anonymous reviewers for the insightful comments.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

D. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan.

Stefan Bordag. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In *EACL*. Citeseer.

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

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Ping Chen, Wei Ding, Chris Bowes, and David Brown. 2009. A fully unsupervised word sense disambiguation method using dependency knowledge. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36, Boulder, Colorado, June. 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, Doha, Qatar, October. Association for Computational Linguistics.

Do Kook Choe and Eugene Charniak. 2013. Naive Bayes word sense induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1437, Seattle, Washington, USA, October. Association for Computational Linguistics.

Kartik Goyal and Eduard H Hovy. 2014. Unsupervised word sense induction using distributional statistics. In *COLING*, pages 1302–1310.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Eric Huang, Richard Socher, Christopher Manning, and Andrew 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 (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July. Association for Computational Linguistics.

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, Denver, Colorado, May–June. Association for Computational Linguistics.

Ioannis Korkontzelos and Suresh Manandhar. 2010. Uoy: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 355–358, Uppsala, Sweden, July. Association for Computational Linguistics.

Mikael Kågebäck, Fredrik Johansson, Richard Johansson, and Devdatt Dubhashi. 2015. Neural context embeddings for automatic discovery of word senses. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 25–32, Denver, Colorado, June. Association for Computational Linguistics.

Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France, April. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal, September. Association for Computational Linguistics.

Suresh Manandhar, Ioannis P Klapaftis, Dmitriy Dligach, and Sameer S Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 63–68. Association for Computational Linguistics.

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

Roberto Navigli and Giuseppe Crisafulli. 2010. Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 116–126, Cambridge, MA, October. Association for Computational Linguistics.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, October. Association for Computational Linguistics.

Ted Pedersen. 2010. Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 363–366, Uppsala, Sweden, July. Association for Computational Linguistics.

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, Doha, Qatar, October. Association for Computational Linguistics.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In Hwee Tou Ng and Ellen Riloff, editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 41–48, Boston, Massachusetts, USA, May 6 - May 7. Association for Computational Linguistics.

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, Los Angeles, California, June. Association for Computational Linguistics.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. 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)*, pages 1793–1803, Beijing, China, July. Association for Computational Linguistics.

Cyrus Shaoul and Chris Westbury. 2010. The westbury lab wikipedia corpus.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

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, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Tim Van de Cruys and Marianna Apidianaki. 2011. Latent semantic word sense induction and disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1476–1485, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jing Wang, Mohit Bansal, Kevin Gimpel, Brian Ziebart, and Clement Yu. 2015. A sense-topic model for word sense induction with unsupervised data enrichment. *Transactions of the Association for Computational Linguistics*, 3:59–71.

Deyi Xiong and Min Zhang. 2014. A sense-based translation model for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1459–1469, Baltimore, Maryland, June. Association for Computational Linguistics.

Xuchen Yao and Benjamin Van Durme. 2011. Nonparametric bayesian word sense induction. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 10–14. Association for Computational Linguistics.