Context-Dependent Sense Embedding*

Lin Qiu[†] and **Kewei Tu**[‡] and **Yong Yu**[†]
[†] Shanghai Jiao Tong University, Shanghai, China, {lqiu,yyu}@apex.sjtu.edu.cn
[‡] ShanghaiTech University, Shanghai, China, tukw@shanghaitech.edu.cn

Abstract

Word embedding has been widely studied and proven helpful in solving many natural language processing tasks. However, the ambiguity of natural language is always a problem on learning high quality word embeddings. A possible solution is sense embedding which trains embedding for each sense of words instead of each word. Some recent work on sense embedding uses context clustering methods to determine the senses of words, which is heuristic in nature. Other work creates a probabilistic model and performs word sense disambiguation and sense embedding iteratively. However, most of the previous work has the problems of learning sense embeddings based on imperfect word embeddings as well as ignoring the dependency between sense choices of neighboring words. In this paper, we propose a novel probabilistic model for sense embedding that is not based on problematic word embedding of polysemous words and takes into account the dependency between sense choices. Based on our model, we derive a dynamic programming inference algorithm and an Expectation-Maximization style unsupervised learning algorithm. The empirical studies show that our model outperforms the state-of-the-art model on a word sense induction task by a 13% relative gain.

1 Introduction

Distributed representation of words (aka word embedding) aims to learn continuous-valued vectors to represent words based on their context in a large corpus. They can serve as input features for algorithms of natural language processing (NLP) tasks. High quality word embeddings have been proven helpful in many NLP tasks (Collobert and Weston, 2008; Turian et al., 2010; Collobert et al., 2011; Maas et al., 2011; Chen and Manning, 2014). Recently, with the development of deep learning, many novel neural network architectures are proposed for training high quality word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b).

However, since natural language is intrinsically ambiguous, learning one vector for each word may not cover all the senses of the word. In the case of a multi-sense word, the learned vector will be around the average of all the senses of the word in the embedding space, and therefore may not be a good representation of any of the senses. A possible solution is sense embedding which trains a vector for each sense of a word. There are two key steps in training sense embeddings. First, we need to perform word sense disambiguation (WSD) or word sense induction (WSI) to determine the senses of words in the training corpus. Then, we need to train embedding vectors for word senses according to their contexts.

Early work on sense embedding (Reisinger and Mooney, 2010; Huang et al., 2012; Chen et al., 2014; Neelakantan et al., 2014; Kageback et al., 2015; Li and Jurafsky, 2015) proposes context clustering methods which determine the sense of a word by clustering aggregated embeddings of words in its context. This kind of methods is heuristic in nature and relies on external knowledge from lexicon like WordNet (Miller, 1995).

*The second author was supported by the National Natural Science Foundation of China (61503248).

Recently, sense embedding methods based on complete probabilistic models and well-defined learning objective functions (Tian et al., 2014; Bartunov et al., 2016; Jauhar et al., 2015) become more popular. These methods regard the choice of senses of the words in a sentence as hidden variables. Learning is therefore done with expectation-maximization style algorithms, which alternate between inferring word sense choices in the training corpus and learning sense embeddings.

A common problem with these methods is that they model the sense embedding of each center word dependent on the word embeddings of its context words. As we previously explained, word embedding of a polysemous word is not a good representation and may negatively influence the quality of inference and learning. Furthermore, these methods choose the sense of each word in a sentence independently, ignoring the dependency that may exist between the sense choices of neighboring words. We argue that such dependency is important in word sense disambiguation and therefore helpful in learning sense embeddings. For example, consider the sentence "He cashed a check at the bank". Both "check" and "bank" are ambiguous here. Although the two words hint at banking related senses, the hint is not decisive (as an alternative interpretation, they may represent a check mark at a river bank). Fortunately, "cashed" is not ambiguous and it can help disambiguate "check". However, if we consider a small context window in sense embedding, then "cashed" cannot directly help disambiguate "bank". We need to rely on the dependency between the sense choices of "check" and "bank" to disambiguate "bank".

In this paper, we propose a novel probabilistic model for sense embedding that takes into account the dependency between sense choices of neighboring words. We do not learn any word embeddings in our model and hence avoid the problem with embedding polysemous words discussed above. Our model has a similar structure to a high-order hidden Markov model. It contains a sequence of observable words and latent senses and models the dependency between each word-sense pair and between neighboring senses in the sequence. The energy of neighboring senses can be modeled using existing word embedding approaches such as CBOW and Skip-gram (Mikolov et al., 2013a; Mikolov et al., 2013b). Given the model and a sentence, we can perform exact inference using dynamic programming and get the optimal sense sequence of the sentence. Our model can be learned from an unannotated corpus by optimizing a max-margin objective using an algorithm similar to hard-EM.

Our main contributions are the following:

1. We propose a complete probabilistic model for sense embedding. Unlike previous work, we model the dependency between sense choices of neighboring words and do not learn sense embeddings dependent on problematic word embeddings of polysemous words.

2. Based on our proposed model, we derive an exact inference algorithm and a max-margin learning algorithm which do not rely on external knowledge from any knowledge base or lexicon (except that we determine the numbers of senses of polysemous words according to an existing sense inventory).

3. The performance of our model on contextual word similarity task is competitive with previous work and we obtain a 13% relative gain compared with previous state-of-the-art methods on the word sense induction task of SemEval-2013.

The rest of this paper is organized as follows. We introduce related work in section 2. Section 3 describes our models and algorithms in detail. We present our experiments and results in section 4. In section 5, a conclusion is given.

2 Related Work

Distributed representation of words (aka word embedding) was proposed in 1986 (Hinton, 1986; Rumelhart et al., 1986). In 2003, Bengio et al. (2003) proposed a neural network architecture to train language models which produced word embeddings in the neural network. Mnih and Hinton (2007) replaced the global normalization layer of Bengio's model with a tree-structure to accelerate the training process. Collobert and Weston (2008) introduced a max-margin objective function

to replace the most computationally expensive max-likelihood objective function. Recently proposed Skip-gram model, CBOW model and GloVe model (Mikolov et al., 2013a; Mikolov et al., 2013b; Pennington et al., 2014) were more efficient than traditional models by introducing a log-linear layer and making it possible to train word embeddings with a large scale corpus. With the development of neural network and deep learning techniques, there have been a lot of work based on neural network models to obtain word embedding (Turian et al., 2010; Collobert et al., 2011; Maas et al., 2011; Chen and Manning, 2014). All of them have proven that word embedding is helpful in NLP tasks.

However, the models above assumed that one word has only one vector as its representation which is problematic for polysemous words. Reisinger and Mooney (2010) proposed a method for constructing multiple sense-specific representation vectors for one word by performing word sense disambiguation with context clustering. Huang et al. (2012) further extended this context clustering method and incorporated global context to learn multi-prototype representation vectors. Chen et al. (2014) extended the context clustering method and performed word sense disambiguation according to sense glosses from WordNet (Miller, 1995). Neelakantan et al. (2014) proposed an extension of the Skip-gram model combined with context clustering to estimate the number of senses for each word as well as learn sense embedding vectors. Instead of performing word sense disambiguation tasks, Kageback et al. (2015) proposed the instance-context embedding method based on context clustering to perform word sense induction tasks. Li and Jurafsky (2015) introduced a multi-sense embedding model based on the Chinese Restaurant Process and applied it to several natural language understanding tasks.

Since the context clustering based models are heuristic in nature and rely on external knowledge, recent work tends to create probabilistic models for learning sense embeddings. Tian et al. (2014) proposed a multi-prototype Skip-gram model and designed an Expectation-Maximization (EM) algorithm to do word sense disambiguation and learn sense embedding vectors iteratively. Jauhar et al. (2015) extended the EM training framework and retrofitted embedding vectors to the ontology of WordNet. Bartunov et al. (2016) proposed a non-parametric Bayesian extension of Skip-gram to automatically learn the required numbers of representations for all words and perform word sense induction tasks.

3 Context-Dependent Sense Embedding Model

We propose the context-dependent sense embedding model for training high quality sense embeddings which takes into account the dependency between sense choices of neighboring words. Unlike pervious work, we do not learn any word embeddings in our model and hence avoid the problem with embedding polysemous words discussed previously. In this section, we will introduce our model and describe our inference and learning algorithms.

3.1 Model

We begin with the notation in our model. In a sentence, let w_i be the i^{th} word of the sentence and s_i be the sense of the i^{th} word. $S(w)$ denotes the set of all the senses of word w. We assume that the sets of senses of different words do not overlap. Therefore, in this paper a word sense can be seen as a lexeme of the word (Rothe and Schutze, 2015).

Our model can be represented as a Markov network shown in Figure 1. It is similar to a high-order hidden Markov model. The model contains a sequence of observable words $(w_1, w_2, \ldots)$ and latent senses $(s_1, s_2, \ldots)$. It models the dependency between each word-sense pair and between neighboring senses in the sequence. The energy function is formulated as follows:

$$E(\mathbf{w}, \mathbf{s}) = \sum_i \Big(E_1(w_i, s_i) + E_2(s_{i-k}, \ldots, s_{i+k}) \Big) \tag{1}$$

Here $\mathbf{w} = \{w_i | 1 \leq i \leq l\}$ is the set of words in a sentence with length l and $\mathbf{s} = \{s_i | 1 \leq i \leq l\}$ is the set of senses. The function E_1 models the dependency between a word-sense pair. As we assume that the sets of senses of different words do not overlap, we can formulate E_1 as follows:

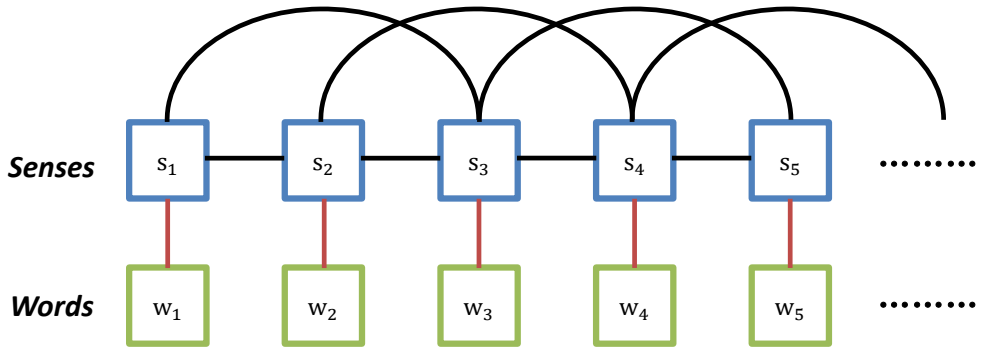

Figure 1: Context-Dependent Sense Embedding Model with window size $k = 1$

$$E_1(w_i, s_i) = \begin{cases} 0 & s_i \in S(w_i) \\ +\infty & s_i \notin S(w_i) \end{cases} \quad (2)$$

Here we assume that all the matched word-sense pairs have the same energy, but it would also be interesting to model the degrees of matching with different energy values in E_1. In Equation 1, the function E_2 models the compatibility of neighboring senses in a context window with fixed size k. Existing embedding approaches like CBOW and Skip-gram (Mikolov et al., 2013a; Mikolov et al., 2013b) can be used here to define E_2. The formulation using CBOW is as follows:

$$E_2(s_{i-k}, \ldots, s_{i+k}) = - \sigma\left(\sum_{i-k \le j \le i+k, j \neq i} V^T(s_j) V'(s_i) \right) \quad (3)$$

Here $V(s)$ and $V'(s)$ are the input and output embedding vectors of sense s. The function σ is an activation function and we use the sigmoid function here in our model. The formulation using Skip-gram can be defined in a similar way:

$$E_2(s_{i-k}, \ldots, s_{i+k}) = - \sum_{i-k \le j \le i+k, j \neq i} \sigma\left(V^T(s_j) V'(s_i) \right) \quad (4)$$

3.2 Inference

In this section, we introduce our inference algorithm. Given the model and a sentence **w**, we want to infer the most likely values of the hidden variables (i.e. the optimal sense sequence of the sentence) that minimize the energy function in Equation 1:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} E(\mathbf{w}, \mathbf{s}) \quad (5)$$

We use dynamic programming to do inference which is similar to the Viterbi algorithm of the hidden Markov model. Specifically, for every valid assignment $A_{i-2k}, \ldots, A_{i-1}$ of every subsequence of senses $s_{i-2k}, \ldots, s_{i-1}$, we define $m(A_{i-2k}, \ldots, A_{i-1})$ as the energy of the best sense sequence up to position $i-1$ that is consistent with the assignment $A_{i-2k}, \ldots, A_{i-1}$. We start with $m(A_1, \ldots, A_{2k}) = 0$ and then recursively compute m in a left-to-right forward process based on the update formula:

$$m(A_{i-2k+1}, \ldots, A_i) = \min_{A_{i-2k}} \Big(m(A_{i-2k}, \ldots, A_{i-1}) + E_1(w_i, A_i) + E_2(A_{i-2k}, \ldots, A_i) \Big) \quad (6)$$

Once we finish the forward process, we can retrieve the best sense sequence with a backward process. The time complexity of the algorithm is $O(n^{4k}l)$ where n is the maximal number of senses of a word. Because most words in a typical sentence have either a single sense or far less than n senses, the actual running time of the algorithm is very fast.

3.3 Learning

In this section, we introduce our unsupervised learning algorithm. In learning, we want to learn all the

input and output sense embedding vectors that optimize the following max-margin objective function:

$$\Theta^* = \arg\min_{\Theta} \sum_{\mathbf{w}\in C} \min_{\mathbf{s}} \sum_{i=1}^{\|\mathbf{w}\|} \sum_{s_{neg}\in S_{neg}(w_i)} \max\Big(1 + E_1(w_i, s_i) + E_2(s_{i-k}, \ldots, s_{i+k}) - E_2(s_{i-k}, \ldots, s_{i-1}, s_{neg}, s_{i+1}, \ldots, s_{i+k}), 0\Big) \tag{7}$$

Here Θ is the set of all the parameters including V and V' for all the senses. C is the set of training sentences. Our learning objective is similar to the negative sampling and max-margin objective proposed for word embedding (Collobert and Weston, 2008). $S_{neg}(w_i)$ denotes the set of negative samples of senses of word w_i which is defined with the following strategy. For a polysemous word w_i, $S_{neg}(w_i) = S(w_i)\backslash\{s_i\}$. For the other words with a single sense, $S_{neg}(w_i)$ is a set of randomly selected senses of a fixed size.

The objective in Equation 7 can be optimized by coordinate descent which in our case is equivalent to the hard Expectation-Maximization algorithm. In the hard E step, we run the inference algorithm using the current model parameters to get the optimal sense sequences of the training sentences. In the M step, with the sense sequences **s** of all the sentences fixed, we learn sense embedding vectors. Assume we use the CBOW model for E_2 (Equation 3), then the M-step objective function is as follows:

$$\Theta^* = \arg\min_{\Theta} \sum_{\mathbf{w}\in C} \sum_{i=1}^{\|\mathbf{w}\|} \sum_{s_{neg}\in S_{neg}(w_i)} \max\Big(1 - \sigma(\sum_{i-k\le j\le i+k, j\neq i} V(s_j)^T V'(s_i)) + \sigma(\sum_{i-k\le j\le i+k, j\neq i} V(s_j)^T V'(s_{neg})), 0\Big) \tag{8}$$

Here E_1 is omitted because the sense sequences produced from the E-step always have zero E_1 value. Similarly, if we use the Skip-gram model for E_2 (Equation 4), then the M-step objective function is:

$$\Theta^* = \arg\min_{\Theta} \sum_{\mathbf{w}\in C} \sum_{i=1}^{\|\mathbf{w}\|} \sum_{i-k\le j\le i+k, j\neq i} \sum_{s_{neg}\in S_{neg}(w_i)} \max\Big(1 - \sigma(V(s_j)^T V'(s_i)) + \sigma(V(s_j)^T V'(s_{neg})), 0\Big) \tag{9}$$

We optimize the M-step objective function using stochastic gradient descent.

We use a mini batch version of the hard EM algorithm. For each sentence in the training corpus, we run E-step to infer its sense sequence and then immediately run M-step (for 1 iteration of stochastic gradient descent) to update the model parameters based on the senses in the sentence. Therefore, the batch size of our algorithm depends on the length of each sentence.

The advantage of using mini batch is twofold. First, while our learning objective is highly non-convex (Tian et al., 2014), the randomness in mini batch hard EM may help us avoid trapping into local optima. Second, the model parameters are updated more frequently in mini batch hard EM, resulting in faster convergence.

Note that before running hard-EM, we need to determine, for each word w, the size of $S(w)$. In our experiments, we used the sense inventory provided by Coarse-Grained English All-Words Task of SemEval-2007 Task 07 (Navigli et al., 2007) to determine the number of senses for each word. The sense inventory is a coarse version of WordNet sense inventory. We do not use the WordNet sense inventory because the senses in WordNet are too fine-grained and are difficult to recognize even for human annotators (Edmonds and Kilgarriff, 2002). Since we do not link our learned senses with external sense inventories, our approach can be seen as performing WSI instead of WSD.

4 Experiments

This section presents our experiments and results. First, we describe our experimental setup including the training corpus and the model configuration.

Word	Nearest Neigbors
bank_1	banking, lender, loan
bank_2	river, canal, basin
bank_3	slope, tilted, slant
apple_1	macintosh, imac, blackberry
apple_2	peach, cherry, pie
date_1	birthdate, birth, day
date_2	appointment, meet, dinner
fox_1	cbs, abc, nbc
fox_2	wolf, deer, rabbit

Table 1: The nearest neighbors of senses of polysemous words

Then, we perform a qualitative evaluation on our model by presenting the nearest neighbors of senses of some polysemous words. Finally, we introduce two different tasks and show the experimental results on these tasks respectively.

4.1 Experimental Setup

4.1.1 Training Corpus

Our training corpus is the commonly used Wikipedia corpus. We dumped the October 2015 snapshot of the Wikipedia corpus which contains 3.6 million articles. In our experiments, we removed the infrequent words with less than 20 occurrences and the training corpus contains 1.3 billion tokens.

4.1.2 Configuration

In our experiments, we set the context window size to 5 (5 words before and after the center word). The embedding vector size is set to 300. The size of negative sample sets of single-sense words is set to 5. We trained our model using AdaGrad stochastic gradient decent (Duchi et al., 2010) with initial learning rate set to 0.025. Our configuration is similar to that of previous work.

Similar to Word2vec, we initialized our model by randomizing the sense embedding vectors. The number of senses of all the words is determined with the sense inventory provided by Coarse-Grained English All-Words Task of SemEval-2007 Task 07 (Navigli et al., 2007) as we explained in section 3.3.

4.2 Case Study

In this section, we give a qualitative evaluation of our model by presenting the nearest neighbors of the senses of some polysemous words. Table 1 shows the results of our qualitative evaluation. We list several polysemous words in the table, and for each word, some typical senses of the word are picked. The nearest neighbors of each sense are listed aside. We used the cosine distance to calculate the distance between sense embedding vectors and find the nearest neighbors.

In Table 1, we can observe that our model produces good senses for polysemous words. For example, the word "bank" can be seen to have three different sense embedding vectors. The first one means the financial institution. The second one means the sloping land beside water. The third one means the action of tipping laterally.

4.3 Word Similarity in Context

This section gives a quantitative evaluation of our model on word similarity tasks. Word similarity tasks evaluate a model's performance with the Spearman's rank correlation between the similarity scores of pairs of words given by the model and the manual labels. However, traditional word similarity tasks like Wordsim-353 (Finkelstein et al., 2001) are not suitable for evaluating sense embedding models because these datasets do not include enough ambiguous words and there is no context information for the models to infer and disambiguate the senses of the words. To overcome this issue, Huang et al. (2012) released a new dataset named Stanford's Contextual Word Similarities (SCWS) dataset. The dataset consists of 2003 pairs of words along with human labelled similarity scores and the sentences containing these words.

Given a pair of words and their contexts, we can perform inference using our model to disambiguate the questioned words. A similarity score can be calculated with the cosine distance between the two embedding vectors of the inferred senses of the questioned words. We also propose another method for calculating similarity scores. In the inference process, we compute the energy of each sense choice of the questioned word and consider the negative energy as the confidence of the sense choice. Then we calculate the cosine similarity between all pairs of senses of the questioned words and compute the average of similarity weighted by the confidence of the senses. The first method is named HardSim and the

Model	Similarity Metrics	$\rho \times 100$
Huang	AvgSim	62.8
Huang	AvgSimC	65.7
Chen	AvgSim	66.2
Chen	AvgSimC	68.9
Neelakantan	AvgSim	67.2
Neelakantan	AvgSimC	69.2
Li		**69.7**
Tian	Model_M	63.6
Tian	Model_W	65.4
Bartunov	AvgSimC	61.2
Ours + CBOW	HardSim	64.3
Ours + CBOW	SoftSim	65.6
Ours + Skip-gram	HardSim	64.9
Ours + Skip-gram	SoftSim	**66.1**

Table 2: Spearman's rank correlation results on the SCWS dataset

second method is named SoftSim.

Table 2 shows the results of our context-dependent sense embedding models on the SCWS dataset. In this table, ρ refers to the Spearman's rank correlation and a higher value of ρ indicates better performance. The baseline performances are from Huang et al. (2012), Chen et al. (2014), Neelakantan et al. (2014), Li and Jurafsky (2015), Tian et al. (2014) and Bartunov et al. (2016). Here Ours + CBOW denotes our model with a CBOW based energy function and Ours + Skip-gram denotes our model with a Skip-gram based energy function. The results above the thick line are the models based on context clustering methods and the results below the thick line are the probabilistic models including ours. The similarity metrics of context clustering based models are AvgSim and AvgSimC proposed by Reisinger and Mooney (2010). Tian et al. (2014) propose two metrics Model_M and Model_W which are similar to our HardSim and SoftSim metrics.

From Table 2, we can observe that our model outperforms the other probabilistic models and is not as good as the best context clustering based model. The context clustering based models are overall better than the probabilistic models on this task. A possible reason is that most context clustering based methods make use of more external knowledge than probabilistic models. However, note that Faruqui et al. (2016) presented several problems associated with the evaluation of word vectors on word similarity datasets and pointed out that the use of word similarity tasks for evaluation of word vectors is not sustainable. Bartunov et al. (2016) also suggest that SCWS should be of limited use for evaluating word representation models. Therefore, the results on this task shall be taken with caution. We consider that more realistic natural language processing tasks like word sense induction are better for evaluating sense embedding models.

4.4 Word Sense Induction

In this section, we present an evaluation of our model on the word sense induction (WSI) tasks. The WSI task aims to discover the different meanings for words used in sentences. Unlike a word sense disambiguation (WSD) system, a WSI system does not link the sense annotation results to an existing sense inventory. Instead, it produces its own sense inventory and links the sense annotation results to this sense inventory. Our model can be seen as a WSI system, so we can evaluate our model with WSI tasks.

We used the dataset from task 13 of SemEval-2013 as our evaluation set (Jurgens and Klapaftis, 2013). The dataset contains 4664 instances inflected from one of the 50 lemmas. Both single-sense instances and instances with a graded mixture of senses are included in the dataset. In this paper, we only consider the single sense instances. Jurgens and Klapaftis (2013) propose two fuzzy measures named Fuzzy B-Cubed (FBC) and Fuzzy Normalized Mutual Information (FNMI) for comparing fuzzy sense assignments from WSI systems. the FBC measure summarizes the performance per instance while the FNMI measure is based on sense clusters rather than instances.

Table 3 shows the results of our context-dependent sense embedding models on this dataset. Here HM is the harmonic mean of FBC and FNMI. The result of AI-KU is from Baskaya et al. (2013), MSSG is from Neelakantan et al. (2014), ICE-online and ICE-kmeans are from Kageback et al. (2015). Our models are denoted in the same way as in the previous section.

From Table 3, we can observe that our models

Model	FBC(%)	FNMI(%)	HM
AI-KU	35.1	4.5	8.0
MSSG	45.9	3.7	6.8
ICE-online	48.7	5.5	9.9
ICE-kmeans	51.1	5.9	10.6
Ours + CBOW	53.8	6.3	11.3
Ours + Skip-gram	56.9	6.7	**12.0**

Table 3: Results of single-sense instances on task 13 of SemEval-2013

outperform the previous state-of-the-art models and achieve a 13% relative gain. It shows that our models can beat context clustering based models on realistic natural language processing tasks.

5 Conclusion

In this paper we propose a novel probabilistic model for learning sense embeddings. Unlike previous work, we do not learn sense embeddings dependent on word embeddings and hence avoid the problem with inaccurate embeddings of polysemous words. Furthermore, we model the dependency between sense choices of neighboring words which can help us disambiguate multiple ambiguous words in a sentence. Based on our model, we derive a dynamic programming inference algorithm and an EM-style unsupervised learning algorithm which do not rely on external knowledge from any knowledge base or lexicon except that we determine the number of senses of polysemous words according to an existing sense inventory. We evaluate our model both qualitatively by case studying and quantitatively with the word similarity task and the word sense induction task. Our model is competitive with previous work on the word similarity task. On the word sense induction task, our model outperforms the state-of-the-art model and achieves a 13% relative gain.

For the future work, we plan to try learning our model with soft EM. Besides, we plan to use shared senses instead of lexemes in our model to improve the generality of our model. Also, we will study unsupervised methods to link the learned senses to existing inventories and to automatically determine the numbers of senses. Finally, we plan to evaluate our model with more NLP tasks.

References

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram.

Osman Baskaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 300–306.

Yoshua Bengio, Holger Schwenk, Jean Sbastien Sencal, Frderic Morin, and Jean Luc Gauvain. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(6):1137–1155.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

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

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.

John Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):257–269.

Philip Edmonds and Adam Kilgarriff. 2002. Introduction to the special issue on evaluating word sense disambiguation systems. *Natural Language Engineering*, 8(4):279–291.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of international conference on World Wide Web*, pages 406–414.

G. E. Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*.

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.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proc. NAACL*, pages 683–693.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299.

Mikael Kageback, Fredrik Johansson, Richard Johansson, and Devdatt Dubhashi. 2015. Neural context embeddings for automatic discovery of word senses. In *Proceedings of NAACL-HLT*, pages 25–32.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *EMNLP*, pages 1722–1732. Association for Computational Linguistics.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *In Workshop at ICLR*.

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.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 641–648.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: coarse-grained english all-words task. In *International Workshop on Semantic Evaluations*, pages 30–35.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*, pages 1059–1069. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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 Schutze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1793–1803. Association for Computational Linguistics.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representation by back-propagating errors. *Nature*, 323(6088):533–536.

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 *COLING*, pages 151–160.

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.