Improving Distributed Representation of Word Sense via WordNet Gloss Composition and Context Clustering

Tao Chen[1] **Ruifeng Xu**[1*] **Yulan He**[2] **Xuan Wang**[1]
[1]Shenzhen Engineering Laboratory of Performance Robots at Digital Stage,
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
[2]School of Engineering and Applied Science, Aston University, UK
chentao1999@gmail.com {xuruifeng,wangxuan}@hitsz.edu.cn
y.he9@aston.ac.uk

Abstract

In recent years, there has been an increasing interest in learning a distributed representation of word sense. Traditional context clustering based models usually require careful tuning of model parameters, and typically perform worse on infrequent word senses. This paper presents a novel approach which addresses these limitations by first initializing the word sense embeddings through learning sentence-level embeddings from WordNet glosses using a convolutional neural networks. The initialized word sense embeddings are used by a context clustering based model to generate the distributed representations of word senses. Our learned representations outperform the publicly available embeddings on 2 out of 4 metrics in the word similarity task, and 6 out of 13 sub tasks in the analogical reasoning task.

1 Introduction

With the rapid development of deep neural networks and parallel computing, distributed representation of knowledge attracts much research interest. Models for learning distributed representations of knowledge have been proposed at different granularity level, including word sense level (Huang et al., 2012; Chen et al., 2014; Neelakantan et al., 2014; Tian et al., 2014; Guo et al., 2014), word level (Rummelhart, 1986; Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2009; Mikolov et al., 2010; Mikolov et al., 2013), phrase level (Socher et al., 2010; Zhang et al., 2014; Cho et al., 2014), sentence level (Mikolov et al., 2010; Socher et al., 2013; Kalchbrenner et al., 2014; Kim, 2014; Le and Mikolov, 2014), discourse level (Ji and Eisenstein, 2014) and document level (Le and Mikolov, 2014).

In distributed representations of word senses, each word sense is usually represented by a dense and real-valued vector in a low-dimensional space which captures the contextual semantic information. Most existing approaches adopted a cluster-based paradigm, which produces different sense vectors for each polysemy or homonymy through clustering the context of a target word. However, this paradigm usually has two limitations: (1) The performance of these approaches is sensitive to the clustering algorithm which requires the setting of the sense number for each word. For example, Neelakantan et al. (2014) proposed two clustering based model: the Multi-Sense Skip-Gram (MSSG) model and Non-Parametric Multi-Sense Skip-Gram (NP-MSSG) model. MSSG assumes each word has the same k-sense (e.g. $k = 3$), i.e., the same number of possible senses. However, the number of senses in WordNet (Miller, 1995) varies from 1 such as "*ben*" to 75 such as "*break*". As such, fixing the number of senses for all words would result in poor representations. NP-MSSG can learn the number of senses for each word directly from data. But it requires a tuning of a hyperparameter λ which controls the creation of cluster centroids during training. Different λ needs to be tuned for different datasets. (2) The initial value of sense representation is critical for most statistical clustering based approaches. However, previous approaches usually adopted random initialization (Neelakantan et al., 2014) or the mean average of candidate words in a gloss (Chen et al., 2014). As a result, they may not produce optimal clustering results for word senses.

Focusing on the aforementioned two problems, this paper proposes to learn distributed representations of word senses through WordNet gloss composition and context clustering. The basic idea is that a word sense is represented as a synonym set (*synset*) in WordNet. In this way, instead of assigning a fixed sense number to each word as in the

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 15–20,
Beijing, China, July 26-31, 2015.

previous methods, different word will be assigned with different number of senses based on their corresponding entries in WordNet. Moreover, we notice that each synset has a textual definition (named as *gloss*). Naturally, we use a convolutional neural network (CNN) to learn distributed representations of these glosses (a.k.a. sense vectors) through sentence composition. Then, we modify MSSG for context clustering by initializing the sense vectors with the representations learned by our CNN-based sentence composition model. We expect that word sense vectors initialized in this way would potentially lead to better representations of word senses generated from context clustering.

The obtained word sense representations are evaluated on two tasks. One is word similarity task, the other is analogical reasoning task provided by WordRep (Gao et al., 2014). The results show that our approach attains comparable performance on learning distributed representations of word senses. In specific, our learned representation outperforms publicly available embeddings on the globalSim and localSim metrics in word similarity task, and 6 in 13 subtasks in the analogical reasoning task.

2 Our Approach

Our proposed approach first train a Continuous Bag-Of-Words (CBOW) model (Mikolov et al., 2013) from a large collection of raw text to generate word embeddings. These word embeddings are then used by a *Sentence Composition Model*, which takes glosses in WordNet as positive training data and randomly replaces part of the sentences as negative training data to construct the corresponding word sense vectors based on a one-dimensional CNN. For example, a WordNet gloss of word *star* is "*an actor who plays a principal role*". This is taken as a positive training example when learning the word sense vector for "*star*". We concatenate the word embedding generated by the CBOW model for each of the words in the gloss, take the concatenated word embeddings as an input to CNN, and get the output vector as one sense vector of word *star*.

The learned sense vectors are fed into a variant of the previously proposed *Multi-Sense Skip-Gram Model* (MSSG) to generates distributed representations of word senses from a text corpus. We name our approach as CNN-VMSSG.

2.1 Training Sense Vectors From WordNet Glosses Using CNN

In this step, we learn the distributed representation of each gloss sentence as the representation of the corresponding synset. The training objective is to minimize the ranking loss below:

$$G_s = \sum_{s \in P} \max\{0, 1 - f(s) + f(s')\} \quad (1)$$

Given a gloss sentence s as a positive training sample, we randomly replace some words (controlled by a parameter λ) in s to construct a negative training sample s'. We compute the scores $f(s)$ and $f(s')$ where $f(\cdot)$ is the scoring function representing the whole CNN architecture without the softmax layer. We expect $f(s)$ and $f(s')$ to be close to 1 and 0 respectively, and $f(s)$ to be larger than $f(s')$ by a margin of 1 for all the sentence in positive training set P.

The CNN architecture used in this component follows the architecture proposed by (Kim, 2014)[1] which is a slight variant of the architecture proposed by (Collobert and Weston, 2008)[2]. It takes a gloss matrix s as input where each column corresponds to the distributed representation $v_{w_i} \in \mathbb{R}^d$ of a word w_i in the sentence.

The idea behind the one-dimensional convolution is to take the dot product of the vector w with each n-gram in the sentence to obtain another sequence c, where n is the width of filter in the convolutional layer. In order to make c to cover different words in the negative sample corresponding a positive sample, in this work, we randomly replace half of the words in a positive training sample to construct a negative training sample ($\lambda = 0.5$). For example, take the WordNet gloss "*an actor who plays a principal role*" as a positive sample, a negative training sample constructed by this method may be "x_1 *actor who* x_2 x_3 *principal* x_4", where x_1 to x_4 are randomly selected words in a vocabulary collected from a large corpus.

In the pooling layer, a max-overtime pooling operation (Collobert et al., 2011), which forces the network to capture the most useful local features produced by the convolutional layers, is applied. The model uses multiple filters (with varying window sizes) to obtain multiple features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is

[1] https://github.com/yoonkim/CNN_sentence
[2] http://ronan.collobert.com/senna/

the probability distribution over labels. The training error propagates back to fine-tune the parameters of the CNN and the input word vectors. The vector generated in the penultimate layer of the CNN architecture is regarded as the sense vector which captures the semantic content of the input gloss to a certain degree.

2.2 Context Clustering and VMSSG Model

Neelakantan et al. (2014) proposed the MSSG model which extends the skip-gram model to learn multi-prototype word embeddings by clustering the word embeddings of context words around each word. In this model, for each word w, the corresponding word embedding $v_w \in \mathbb{R}^d$, k-sense vector $v_{s_k} \in \mathbb{R}^d$ $(k = 1, 2, \ldots, K)$ and k-context cluster with center $\mu_k \in \mathbb{R}^d$ $(k = 1, 2, \ldots, K)$ are initialized randomly. The sense number K of each word is a fixed parameter in the training algorithm.

We improve the MSSG model by using the learned CBOW word embedding to initialize v_w and the sense vector trained by the sentence composition model to initialize v_{s_k}. We also use the sense number of each word in WordNet K_w to replace K. We named this model as a variant of the MSSG (VMSSG) model.

Algorithm 1 Algorithm of VMSSG model

Input: D, d,$K_1, ..., K_w, ..., K_{|V|}$, M.
Initialize: $\forall w \in V, k \in \{1, \ldots, K_w\}$, initialize v_w to a pre-trained word vector, $v_{s_k^w}$ to a pre-trained sense vector for word w with sense k, and μ_k^w to a vector of random real value $\in (-1, 1)^d$.
for each w in D **do**
  $r \leftarrow$ random number $\in [1, M]$
  $C \leftarrow \{w_{i-r}, ..., w_{i-1}, w_{i+1}, ..., w_{i+r}\}$
  $v_c \leftarrow \frac{1}{2\times r} \sum_{w \in C} v_w$
  $\hat{k} = \arg\max_k \{\text{sim}(\mu_k^w, v_c)\}$
  Assign C to context cluster $\hat{k}$.
  Update $\mu_{\hat{k}}$.
  $C' = \text{NoisySamples}(C)$
  Gradient update on $v_{s_k^w}$, v_w in C, C'.
end for
Output: $v_{s_k^w}$, v_w, $\forall w \in V, k \in \{1, \ldots, K_w\}$

The training algorithm of the VMSSG model is shown as Algorithm 1, where D is a text corpus, V is the vocabulary of D, $|V|$ is the vocabulary size, M is the size of context window, v_w is the word embedding for w, s_k^w is a kth context cluster of word w, μ_k^w is the centroid of cluster k for word w. The function NoisySamples(C) randomly replaces context words with noisy words from V.

3 Evaluation and Discussion

3.1 Experimental Setup

In all experiments, we train word vectors and sense vectors on a snapshot of Wikipedia in April 2010[3] (Shaoul, 2010), previously used in (Huang et al., 2012; Neelakantan et al., 2014). WordNet 3.1 is used for training the sentence composition model. A publicly available word vectors trained by CBOW from Google News[4] are used as pre-trained word vectors for CNN.

For training CNN, we use: rectified linear units, filter windows of 3, 4, 5 with 100 feature maps each, AdaDelta decay parameter of 0.95, the dropout rate of 0.5. For training VMSSG, we use *MSSG-KMeans* as the clustering algorithm, and CBOW for learning sense vectors. We set the size of word vectors to 300, using boot vectors and sense vectors. For other parameter, we use default parameter settings for MSSG.

3.2 Word Similarity Task

We evaluate our embeddings on the Contextual Word Similarities (SCWS) dataset (Huang et al., 2012). It contains 2,003 pairs of words and their sentential contexts. Each pair is associated with 10 to 16 human judgments of similarity on a scale from 0 to 10. We use the same metrics in (Neelakantan et al., 2014) to measure the similarity between two words given their respective context. The *avgSim* metric computes the average similarity of all pairs of prototype vectors for each word, ignoring context. The *avgSimC* metric weights each similarity term in avgSim by the likelihood of the word context appearing in its respective cluster. The *globalSim* metric computes each word vector ignoring senses. The *localSim* metric chooses the most similar sense in context to estimate the similarity of a words pair.

We report the Spearman's correlation $\rho \times 100$ between a model's similarity scores and the human judgments in Table 1.[5]

[3] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[4] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

[5] The localSim metric of *Unified-WSR* is not reported in (Chen et al., 2014).

Model	avgSim	avgSimC	globalSim	localSim
Huang et al. 50d	62.8	65.7	58.6	26.1
Unified-WSR 200d	66.2	68.9	64.2	-
MSSG 300d	67.2	**69.3**	65.3	57.3
NP-MSSG 300d	**67.3**	69.1	65.5	59.8
CNN-VMSSG 300d	65.7	66.4	**66.3**	**61.1**

Table 1: Experimental results in the SCWS task.

Subtask	Word Pairs	C&W	CBOW	MSSG	NP-MSSG	CNN-VMSSG
Antonym	973	0.28	**4.57**	0.25	0.10	1.01
Attribute	184	0.22	1.18	0.03	0.15	**1.63**
Causes	26	0.00	1.08	0.31	0.31	**1.23**
DerivedFrom	6,119	0.05	**0.63**	0.09	0.05	0.17
Entails	114	0.05	0.38	0.49	0.34	**1.29**
HasContext	1,149	0.12	0.35	**1.73**	1.56	1.41
InstanceOf	1,314	0.08	0.58	**2.52**	2.34	2.46
IsA	10,615	0.07	0.67	0.15	0.08	**0.86**
MadeOf	63	0.03	0.72	0.80	0.48	**1.28**
MemberOf	406	0.08	**1.06**	0.14	0.86	0.90
PartOf	1,029	0.31	1.27	**1.50**	0.73	0.48
RelatedTo	102	0.00	0.05	0.12	0.11	**1.28**
SimilarTo	3,489	0.02	**0.29**	0.03	0.01	0.12

Table 2: Experimental results in the analogical reasoning task.

It is observed that our model achieves the best performance on the *globalSim* and *localSim* metrics. It indicates that the use of pre-trained word vectors and initializing sense vectors with the embeddings learned from WordNet glosses are indeed helpful in improving the quality of both global word vectors and sense-level word vectors. Our approach performs worse on *avgSim* and *avgSimC*. One possible reason is that we set the number of context clusters for each word to be the same as the number of its corresponding senses in WordNet. However, not all senses appear in the our experimented corpus which could lead to fragmented context clustering results. One possible way to alleviate this problem is to perform post-processing to merge clusters which have smaller inter-cluster differences or to remove sense clusters which are under-represented in our data. We will leave it as our future work.

3.3 Analogical Reasoning Task

The analogical reasoning task introduced by (Mikolov et al., 2013) consists of questions of the form "a is to b is as c is to _", where (a, b) and (c, _) are two word pairs. The goal is to find a word d^* in vocabulary V whose representation vector is the closest to $v_b - v_a + v_c$.

WordRep is a benchmark collection for the research on learning distributed word representations, which expands the Mikolov et al.'s analogical reasoning questions. In our experiments, we use one evaluation set in WordRep, the WordNet collection which consists of 13 sub tasks.

We use the precision $p \times 100$ as metric for each sub task. Table 2 shows the results on the 13 sub tasks. The *Word Pair* column is the number of word pairs of each sub task. The results of C&W were obtained using the 50-dimensional word embeddings that were made publicly available by Turian et al. (2010).[6] The CBOW results were previously reported in (Gao et al., 2014).

It can be observed that among 13 subtasks, our model outperforms the others by a good margin in 6 subtasks, *Attribute*, *Causes*, *Entails*, *IsA*, *MadeOf* and *RelatedTo*.

3.4 Discussion

Although our evaluation results on the word similarity task and the analogical reasoning task show that our proposed approach outperforms a number of existing word representation methods in some

[6] http://metaoptimize.com/projects/wordreprs/

of the subtasks, it is worth noting that both tasks do not consider the full spectrum of senses. In specific, the analogical reasoning task was originally designed for evaluating single-prototype word representations which ignore that a word could have multiple meanings. Compared to single-prototype word vectors, evaluating sense vectors requires a significantly larger search space since each word could be represented by multiple sense vectors depending on the context. One may also argue that the analogical reasoning task may not be the most appropriate one in evaluating multiple-prototype word vectors since the context information is not available. In the future, we plan to evaluate our learned multiple-prototype word vectors in more relevant NLP tasks such as word sense disambiguation and question answering.

Our proposed approach initializes sense vectors using the learned sentence embeddings from WordNet glosses. In other low resourced languages, it is still possible to intialize sense vectors based on, for example, the word meanings found in language-specific dictionaries.

4 Conclusion and Future Work

This paper presents a method of incorporating WordNet glosses composition and context clustering based model for learning distributed representations of word senses. By initializing sense vectors using the embeddings learned by a sentence composition from WordNet glosses, the context clustering method is able to generate better distributed representations of word senses. The obtained word sense representations achieve state-of-the-art results on the globalSim and localSim metrics in the word similarity task and in 6 sub tasks of the analogical reasoning task. It shows the effectiveness of our proposed learning algorithm for generating word sense distributed representations.

Considering the coverage of word senses in our training data, in future work we plan to filter out those sense vectors which are under-represented in the training corpus. We will also further investigate the feasibility of applying the multi-prototype word embeddings in a wide range of NLP tasks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61370165, 61203378), National 863 Program of China 2015AA015405, the Natural Science Foundation of Guangdong Province (No. S2013010014475), Shenzhen Development and Reform Commission Grant No.[2014]1507, Shenzhen Peacock Plan Research Grant KQCX20140521144507925 and Baidu Collaborate Research Funding.

References

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

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.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

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 (ICML)*, 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.

Bin Gao, Jiang Bian, and Tie-Yan Liu. 2014. Wordrep: A benchmark for research on learning word representations. In *ICML 2014 Workshop on Knowledge-Powered Deep Learning for Text Mining (KPDLTM2014)*.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 497–507.

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 (ACL)*, pages 873–882. Association for Computational Linguistics.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13–24, Baltimore, Maryland, June. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, October.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 1188–1196.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*, pages 1045–1048.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations (ICLR)*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.

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.

DE Rummelhart. 1986. Learning representations by back-propagating errors. *Nature*, 323(9):533–536.

Cyrus Shaoul. 2010. The westbury lab wikipedia corpus. *Edmonton, AB: University of Alberta*.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

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 the 25th International Conference on Computational Linguistics (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 (ACL)*, pages 384–394.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 111–121, Baltimore, Maryland, June. Association for Computational Linguistics.