Online Learning of Interpretable Word Embeddings

Hongyin Luo[1], **Zhiyuan Liu**[1,2] *, **Huanbo Luan**[1], **Maosong Sun**[1,2]

[1] Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems, National Lab for Information Science and Technology, Tsinghua University, Beijing, China

[2] Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China

Abstract

Word embeddings encode semantic meanings of words into low-dimension word vectors. In most word embeddings, one cannot interpret the meanings of specific dimensions of those word vectors. Non-negative matrix factorization (NMF) has been proposed to learn interpretable word embeddings via non-negative constraints. However, NMF methods suffer from scale and memory issue because they have to maintain a global matrix for learning. To alleviate this challenge, we propose online learning of interpretable word embeddings from streaming text data. Experiments show that our model consistently outperforms the state-of-the-art word embedding methods in both representation ability and interpretability. The source code of this paper can be obtained from `http://github.com/skTim/OIWE`.

1 Introduction

Word embeddings (Turian et al., 2010) aim to encode semantic meanings of words into low-dimensional dense vectors. As compared with traditional one-hot representation and distributional representation, word embeddings can better address the sparsity issue and have achieved success in many NLP applications recent years.

There are two typical approaches for word embeddings. The neural-network (NN) approach (Bengio et al., 2006) employs neural-based techniques to learn word embeddings. The matrix factorization (MF) approach (Pennington et al., 2014) builds word embeddings by factorizing word-context co-occurrence matrices. The MF approach requires a global statistical matrix, while the NN approach can flexibly perform learning from streaming text data, which is efficient in both computation and memory. For example, two recent NN methods, Skip-Gram and Continuous Bag-of-Word Model (CBOW) (Mikolov et al., 2013a; Mikolov et al., 2013b), have achieved impressive impact due to their simplicity and efficiency.

For most word embedding methods, a critical issue is that, we are unaware of what each dimension represent in word embeddings. Hence, the latent dimension for which a word has its largest value is difficult to interpret. This makes word embeddings like a black-box, and prevents them from being human-readable and further manipulation.

People have proposed non-negative matrix factorization (NMF) for word representation, denoted as non-negative sparse embedding (NNSE) (Murphy et al., 2012). NNSE realizes interpretable word embeddings by applying non-negative constraints for word embeddings. Although NNSE learns word embeddings with good interpretabilities, like other MF methods, it also requires a global matrix for learning, thus suffers from heavy memory usage and cannot well deal with streaming text data.

Inspired by the characteristics of NMF methods (Lee and Seung, 1999), we note that, non-negative constraints only allow additive combinations instead of subtractive combinations, and lead to a parts-based representation. Hence, the non-negative constraints derive interpretabilities of word embeddings. In this paper, we aim to design an online NN method to efficiently learn interpretable word embeddings. In order to achieve the goal of interpretable embeddings, we design projected gradient descent (Lin, 2007) for optimization so as to apply non-negative constraints on NN methods such as Skip-Gram. We also employ adaptive gradient descent (Sun et al., 2012) to speedup learning convergence. We name the proposed models as online interpretable word embeddings (OIWE).

*Corresponding author: Z. Liu (liuzy@tsinghua.edu.cn)

For experiments, we implement OIWE based on Skip-Gram. We evaluate the representation performance of word embedding methods on the word similarity computation task. Experiment results show that, our OIWE models are significantly superior to other baselines including Skip-Gram, RNN and NNSE. We also evaluate the interpretability performance on the word intrusion detection task. The results demonstrate the effectiveness of OIWE as compared to NNSE.

2 Our Model

In this section, we first introduce Skip-Gram and then introduce the proposed online interpretable word embeddings based on Skip-Gram.

2.1 Skip-Gram

Skip-Gram (Mikolov et al., 2013b) is simple and effective to learn word embeddings. The objective of Skip-Gram is to make word vectors good at predicting its context words. More specifically, given a word sequence $\{w_1, w_2, \ldots, w_T\}$, Skip-Gram aims to maximize the average log probability

$$\frac{1}{T}\sum_{1}^{T}\Big(\sum_{-k\le j\le k, j\neq 0}\log \Pr(w_{t+j}|w_t)\Big), \quad (1)$$

where k is the context window size, and $\Pr(w_{t+j}|w_t)$ indicates the probability of seeing w_{t+j} in the context of w_t, which are measured with softmax function

$$\Pr(w_{t+j}|w_t) = \frac{\exp\big(\mathbf{w}_{t+j}\cdot\mathbf{w}_t\big)}{\sum_{w\in W}\exp\big(\mathbf{w}\cdot\mathbf{w}_t\big)}, \quad (2)$$

where $\mathbf{w}_{t+j}$ and $\mathbf{w}_t$ are word embeddings of w_{t+j} and w_t, and W is the vocabulary size. Since the computation of full softmax is time consuming, the techniques of hierarchical softmax and negative sampling (Mikolov et al., 2013b) are proposed for approximation.

Take negative sampling for example. The log probability $\Pr(w_{t+j}|w_t)$ can be approximate by

$$\log\sigma\big(\mathbf{w}_{t+j}\cdot\mathbf{w}_t\big) + \sum_{w\in N_t}\log\sigma\big(\mathbf{w}\cdot\mathbf{w}_t\big), \quad (3)$$

where $\sigma(x) = 1/(1+\exp(-x))$, and N_t is the set of negative samples as compared to the corresponding context word $\mathbf{w}_{t+j}$. The task can be regarded as to distinguish the context word $\mathbf{w}_{t+j}$ from negative samples.

For Skip-Gram with negative sampling, we can perform stochastic gradient descent for learning. The update rule for the positive/negative context words $u \in \{w_{t+j}\}\cup N_t$ is

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \gamma\big[I_{w_t}(u) - \sigma(\mathbf{u}\cdot\mathbf{w}_t)\big]\mathbf{w}_t^i, \quad (4)$$

where $I_{w_t}(u) = 1$ when w is the positive context word of w_t and $I_{w_t}(u) = 0$ when w is negative, i is the iteration number, and γ is the learning rate. Correspondingly, the update rule for the input word w_t is

$$\mathbf{w}_t^{i+1} = \mathbf{w}_t^i + \gamma\sum_{u\in\{w_t\}\cup N_t}\big[I_{w_t}(u) - \sigma(\mathbf{u}\cdot\mathbf{w}_t)\big]\mathbf{u}_t^i. \quad (5)$$

We note that, the learning rate γ in Skip-Gram is shared by all word embeddings.

2.2 OIWE

In order to learn interpretable word embeddings, we have to make the word embeddings learned in Skip-Gram keep non-negative. In order to achieve this goal, we have to constrain the update rules in Equation (4) and (5) as follows:

$$\mathbf{x}_k^{i+1} = P\big[\mathbf{x}_k^i + \gamma\nabla f(\mathbf{x}_k)\big], \quad (6)$$

where x may be u or w_t, k is the corresponding dimension in word embedding $\mathbf{x}$, $\nabla f(\mathbf{x}_k)$ indicates the gradient corresponding to $\mathbf{x}_k$, and $P[\cdot]$ is defined as

$$P[x] = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \le 0. \end{cases} \quad (7)$$

Motivated by the projected gradient descent methods for NMF (Lin, 2007), in this paper we propose two methods for Skip-Gram to realize the constraint in Equation (6).

Naive Projected Gradient (NPG). In NPG, we consider the most straightforward update strategy by simply setting

$$\mathbf{x}_k^{i+1} = \max\big(0, \mathbf{x}_k^i + \gamma\nabla f(\mathbf{x}_k)\big). \quad (8)$$

The method has been used for NMF (Lin, 2007) although the details are not discussed.

The NPG method only constrains the violated dimensions without taking the update consistency among dimensions of a word embedding into account. For example, if many dimensions encounter $\mathbf{x}_k^i + \gamma\nabla f(\mathbf{x}_k) < 0$ at the same time, which are set to 0 with Equation (8) with other

dimensions unchanged, the updated word embedding may heavily deviate from its semantic meaning. Hence, NPG may suffer from instable updating results. To address this issue, we propose to employ the following improved projected gradient method.

Improved Projected Gradient (IPG). In order to make the non-negative update more consistent among dimensions, we design an improved projected gradient by iteratively finding the most appropriate learning rate γ. The basic idea is that, we will find a good learning rate γ to make less dimensions violate the non-negative constraint.

More specifically, in Equation (6), for a learning rate γ, we define the *violation ratio* as

$$R(\gamma) = \frac{\left|\{k|\mathbf{x}_k^i > 0, \mathbf{x}_k^i + \gamma \nabla f(\mathbf{x}_k) < 0\}\right|}{K}, \quad (9)$$

where K is the dimension size of word embeddings. The violation ratio indicates how many dimensions violate the non-negative constraint and require to be set to 0. When the learning rate γ decreases, the violation ratio will also decrease, and the zero-setting in Equation (8) will bring less deviation to word embeddings.

We set a threshold δ for the violation ratio $R(\gamma)$ and a lower bound γ_L for the learning rate γ. Starting from an initial learning rate γ^0, we will repeatedly decrease the learning rate by

$$\gamma^{m+1} = \gamma^m \cdot \beta \quad (10)$$

with $0 < \beta < 1$ until

$$R(\gamma^{m+1}) < \delta \quad \text{or} \quad \gamma^{m+1} \leq \gamma_L, \quad (11)$$

and then update with Equation (8) using γ^{m+1}. In nature, the updating constraint of learning rate in Equation (11) play a similar role to Equation (13) in (Lin, 2007), which aims to prevent the projection operation from heavily deviating the word embeddings.

2.3 More Optimization Details

In experiments, we explore many optimization methods and find the following two strategies are important: (1) *Adaptive Gradient Descent*. Following the idea from (Sun et al., 2012), we maintain different learning rates γ_w for each word w, and the learning rates for those high-frequency words may decrease faster than those low-frequency words. This will speedup the convergence of word embedding learning. (2) *Unified Word Embedding Space*. Different from original Skip-Gram (Mikolov et al., 2013b) which learn embeddings of w_t and its context words w_{t+j} in two separate spaces, in this paper both w_t and its context words w_{t+j} share the same embedding space. Hence, a word embedding may get more opportunities for learning.

3 Experiments

In this section, we investigate the representation performance and interpretability of our OIWE models with other baselines including typical N-N and MF methods.

The representation performance is evaluated with the word similarity computation task, and the interpretability is evaluated with the word intrusion detection task. For the both tasks, we train our OIWE models using the `text8` corpus obtained from *word2vec* website[1], and the OIWE models achieve the best performance by setting the dimension number $K = 300$, $\beta = 0.6$, $\delta = 1/60$, and $\gamma_L = 2.5 \times 10^{-6}$.

3.1 Word Similarity Computation

Following the settings in (Murphy et al., 2012), we also select the following three sets for word similarity computation: (1) **WS-203**, the strict-similarity subset of 203 pairs (Agirre et al., 2009) selected from the wordsim-353 (Finkelstein et al., 2001), (2) **RG-65**, 65 concrete word pairs built by (Rubenstein and Goodenough, 1965) and (3) **MEN**, $3,000$ word pairs built by (Bruni et al., 2014). The performance is evaluated with the Spearman coefficient between human judgements and similarities calculated using word embeddings.

We select three baselines including Skip-Gram (Mikolov et al., 2013b), recurrent neural networks (RNN) (Mikolov et al., 2011) and NNSE (Murphy et al., 2012). For Skip-Gram, we report the result we learned using *word2vec* on `text8` corpus. The result of RNN is from (Faruqui and Dyer, 2014) and the one of NNSE is from (Murphy et al., 2012).

The evaluation results of word similarity computation are shown in Table 1. We can observe that: (1) The OIWE models consistently outperform other baselines. (2) IPG generally achieves better representation performance than

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

Model	WS-203	RG-65	MEN
Skip-Gram	67.35	50.49	52.56
RNN	49.28	50.19	43.44
NNSE	51.06	56.48	-
OIWE-NPG	63.71	56.85	**57.60**
OIWE-IPG	**71.74**	**57.16**	56.68

Table 1: Spearman coefficient results (%) on word similarity computation.

NPG. This indicates consistent updates are important for learning of word embeddings. One can refer to `http://github.com/skTim/OIWE` for the evaluation results on more evaluation datasets.

3.2 Word Intrusion Detection

We evaluate interpretability of word embeddings with the task of word intrusion detection proposed by (Murphy et al., 2012). In this task, for each dimension we create a word set containing top-5 words in this dimension, and intruce a noisy word from the bottom half of this dimension which ranks high in other dimensions. Human editors are asked to check each word set and try to pick out the intrusion words, and the detection precision indicates the interpretability of word embedding models. Note that, for this task we do not perform normalization for word vectors.

Model	Precision
Skip-Gram	32.62
NNSE	92.00
OIWE-NPG	61.40
OIWE-IPG	**94.80**

Table 2: Experiment results (%) on word intrusion detection.

The evaluation results are shown in Table 2. We can observe that: (1) Skip-Gram performs poor in word intrusion detection without doubt since it is uninterpretable in nature. (2) The OIWE-NPG model achieves better interpretability as compared to Skip-Gram, but performs much worse than the OIWE-IPG model. The OIWE-IPG model achieves competitive interpretability with NNSE. This indicates that reducing violation rations in word embedding learning is crucial for preserving interpretability.

In Table 3, we show top-5 words for some dimensions, which clearly demonstrate semantic meanings of these dimensions. One can also refer to `http://github.com/skTim/OIWE` to find top-5 words for all dimensions.

No.	Top Words
1	type, form, way, kind, manner
2	translates, describes, combines, included, includes
3	gospel, baptism, jesus, faith, judaism
4	Franz, Johann, Wilhelm, Friedrich, von
25	prominent, famous, important, influential, popular

Table 3: Top words of some dimensions in word embeddings.

3.3 Influence of Dimension Numbers

The dimension number is an important configuration in word embeddings. In Fig. 1 we show the performance of OIWE and Skip-Gram on word similarity computation with varying dimension numbers.

From the figure, we can observe that: (1) The both models achieve their best performance under the same dimension number. This indicates that OIWE, to some extent, inherits the representation power of Skip-Gram. (2) The performance of OIWE seems to be more sensitive to dimension numbers. When the dimension number changes from 300 to 200 or 400, the performance drops much quickly than Skip-Gram. The reason may be as follows. OIWE has to concern about both representation ability of word embeddings and interpretability of each dimension. An appropriate dimension number is critical to make each dimension interpretable, just like the cluster number is important for clustering. On the contrary, Skip-Gram is much free to learn word embeddings only concerning about representation ability. (3) The performance of OIWE with various dimensions also varies on different evaluation datasets. For example, OIWE-IPG with $K = 400$ gets 68.74 on MEN, which is much better than that with $K = 300$. In future work, we will extensively investigate the characteristics of OIWE with respect to dimension numbers and other hyperparameters.

4 Conclusion and Future Work

In this paper, we present online interpretable word embeddings. The OIWE models perform project-

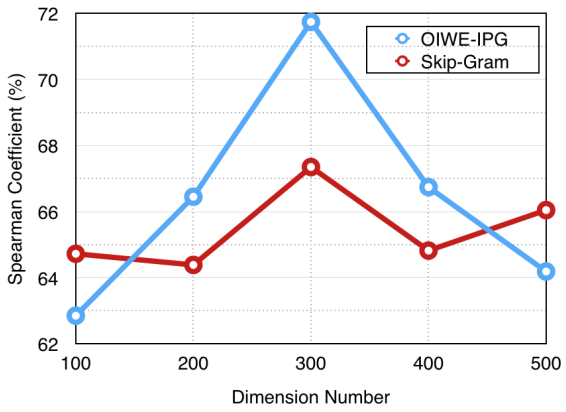

Figure 1: Influence of Dimension Number on Words Similarity

ed gradient descent to apply non-negative constraints on NN methods such as Skip-Gram. Experiment results on word similarity computation and word intrusion detection demonstrate the effectiveness and efficiency of our models in both representation ability and interpretability. We also note that, our models can be easily extended to other NN methods.

In future, we will explore the following research issues: (1) We will extensively investigate the characteristics of OIWE with respect to various hyperparameters including dimension numbers. (2) We will evaluate the performance of our OIWE models in various NLP applications. (3) We will also investigate possible extensions of our OIWE models, including multiple-prototype models for word sense embeddings and semantic compositions for phrase embeddings.

Acknowledgments

Zhiyuan Liu and Maosong Sun are supported by National Key Basic Research Program of China (973 Program 2014CB340500) and National Natural Science Foundation of China (NSFC No. 62102140). Huanbo Luan is supported by the National Natural Science Foundation of China (NSFC No. 61303075). This research is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of HLT-NAACL*, pages 19–27.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *JAIR*, 49:1–47.

Manaal Faruqui and Chris Dyer. 2014. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of ACL System Demonstrations*.

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 WWW*, pages 406–414. ACM.

Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Chuan-bi Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Honza Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of ICASSP*, pages 5528–5531. IEEE.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Brian Murphy, Partha Pratim Talukdar, and Tom M Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING*, pages 1933–1950.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of EMNLP*, 12:1532–1543.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of ACL*, pages 253–262.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.