Refining Word Reesprentations by Manifold Learning

Chu Yonghe, Hongfei Lin*, Liang Yang, Yufeng Diao, Shaowu Zhang and **Fan Xiaochao**
Dalian University of Technology
yhchu@mail.dlut.edu.cn, hflin@dlut.edu.cn, liang@dlut.edu.cn, diaoyufeng@mail.dlut.edu.cn,
zhangsw@dlut.edu.cn, fxc1982@mail.dlut.edu.cn

Abstract

Pre-trained distributed word representations have been proven useful in various natural language processing (NLP) tasks. *However, the effect of words' geometric structure on word representations has not been carefully studied yet. The existing word representations methods underestimate the words whose distances are close in the Euclidean space, while overestimating words with a much greater distance. In this paper, we propose a word vector refinement model to correct the pre-trained word embedding, which brings the similarity of words in Euclidean space closer to word semantics by using manifold learning. This approach is theoretically founded in the metric recovery paradigm. Our word representations have been evaluated on a variety of lexical-level intrinsic tasks (semantic relatedness, semantic similarity) and the experimental results show that the proposed model outperforms several popular word representations approaches.

1 Introduction

In recent years, distributed word representation has received widespread attention from researchers as its ability to capture the syntactic and semantic information of words [Mikolov *et al*., 2013b; Collobert *et al*., 2011; Pennington *et al*., 2014a; Mikolov *et al*., 2013a]. Extensive research has also been devoted to distributed word representation learning, such as literatures [Pennington *et al*., 2014a; Mikolov *et al*., 2013a]. Based on the distributed hypothesis, the above methods use word co-occurrence to map words into low-dimensional dense vectors while maintaining the semantic information of the words. In this low-dimensional vector space, it is convenient to measure the similarity between two words by using the measurement methods, such as distance or angle. Generally, distributed word representation has been founded of widespread application in natural language processing tasks due to its good performance. Many empirical results show that such pre-trained word representations can enhance the performance of supervised models on a variety of NLP tasks, e.g., chunking, named entity recognition, and language modelling [Collobert and Weston, 2008]. Although different word representations have different structures, they all use word co-occurrence to train word vectors in an iterative manner which predict current words by words in the context, such as C&W [Collobert *et al*., 2011] and Continuous Bag-Of-Words (CBOW) [Mikolov *et al*., 2013a]. Another type of word representation methods is based on predicting the adjacent word from the current word, such as Skip-Gram (SG) [Mikolov *et al*., 2013b] and its variants [Qiu *et al*., 2014]. The above two methods of word representation use the local contextual features to train the word vector without considering the global features from the corpus. To address this problem, Pennington *et al*.[2014a]propose the Glove model, which takes into account both local context features and global features of the corpus.

The above intuitions to train word embedding have been proven to be useful. The relationship between the learned word and context representations should be carefully studied in mathematics or geometry. In cognitive psychology, these concepts are assumed to be points in Euclidean space. This view has been confirmed by human judgment experiments. Words are mapped to low-dimensional dense vectors and existed in Euclidean space as points. Thus, semantically similar words have a smaller distance in Euclidean space, whereas opposite words have a larger distance. However, the existing word representation models do not take into account the geometric information between words. As we know, the similarity of words in the Euclidean metric space is inconsistent with human empirical judgment. This is exemplified by the WS353[Finkelstein *et al*., 2001]word similarity ground truth in Figure 1.

Based on the Common Crawl corpus (42B), the Glove model is used to train 300-dimensional word vectors. The similarity between words is measured by the cosine distance, as shown in the following Figure 2.

$$sim(\text{"}shore\text{"},\text{"}woodland\text{"}) = 3.08$$
$$sim(\text{"}physics\text{"},\text{"}proton\text{"}) = 8.12$$

Figure 1: Judging word similarity by human experience.

$$sim(\text{"}shore\text{"},\text{"}woodland\text{"}) = 0.36$$
$$sim(\text{"}physics\text{"},\text{"}proton\text{"}) = 0.33$$

Figure 2: Word similarity in Euclidean metric space.

* This is the corresponding author.

Figure 1 shows the real similarity between "shore" and "woodland" based on human judgment, and the similarity between "physics" and "proton" is also shown. Figure 2 shows the similarity of the word vectors trained by the Glove model in the Euclidean space. However, Figure 1 and Figure 2 show two opposite results that the word vector generally exists in a high-dimensional manifold by exhibiting a nonlinear structure. The traditional word vector metrics treat the observation space of the word vector as high-dimensional space. The word vector to be analyzed and processed is regarded as points distributed in the high-dimensional Euclidean space [Roweis and Saul, 2000], and the distance between the points is thus measured by the straight line distance of the Euclidean geometry. However, it is well known that Euclidean space is a globally linear space, that is, there exists a Cartesian coordinate system defined over the entire space. If the data distribution is globally linear, these methods will be able to effectively learn the linear structure of the data. However, if the word vector distribution is highly nonlinear or strongly attribute-related, it is difficult to obtain the inherent geometry of a nonlinear dataset and its regularity based on the assumption of the global linear structure of Euclidean space. In order to solve the inconsistency between semantic similarity and Euclidean measurement of words, in this paper, manifold learning is introduced into the representation of distributed words. Manifold learning describes the local geometric structure information between sample points of word vectors by constructing the adjacency graph structure of word vectors in high-dimensional space. By tiling the sample distribution group in the high-dimensional feature space to a low-dimensional space, the sample distribution in the original space may be distorted. After tiling, it will be more favorable to measure the distance between word vectors, which can better reflect the similarity between two samples.

In this paper, we study the nature of word representation learning algorithms under a general framework, aiming to establish whether the learned representation of a target word belongs to the conic hull formed by the representations of its contexts. It means that the directions of word representations are strongly correlated with the context representations, while their geometric structure is relatively neglected. Such observation can explain why the similarity between word vectors obtained by the Glove model is inconsistent with human judgment. Inspired by this observation, we explore the possibility of learning the word vectors the geometric structure of which is taken into account by manifold learning. Based on estimating the distance between nearby words, manifold learning is used to direct similarity assignment in a local neighborhood, while the distance between words that are further apart is approximated by multiple neighborhoods by the manifold shape. Manifold learning effectively depicts the geometric structure information of word vectors in high-dimensional space, and significantly improves the word embedding effect of the current word distributed model.

The manifold learning algorithm MLLE [Zhang and Wang, 2006] is applied to the Glove model to obtain the similarity between "shore" and "woodland", as well as "physics" and "proton". The results are shown in Figure 3.

$$sim("shore","woodland") = 0.1$$
$$sim("physics","proton") = 0.23$$

Figure 3: Improved word similarity by manifold learning.

It can be seen from Figure 1−3 that the similarity of the word vectors obtained by using the manifold learning to improve the Glove model, which is consistent with the similarity of the words judged by the human experience given in Figure 1.

In this paper, we use manifold learning to improve the representation of word vectors. Our approach has two major contributions:

- The words' similarity obtained by the current distributed words is inconsistent with that determined by human judgment. In the perspective of manifold learning, we give a reasonable explanation.
- For the influence of the neighborhood point selection on the word representation when manifold learning is applied to word representation, dynamic selection of neighborhood points solves the singularity problem caused by matrix.

In order to verify the validity of the model, we develop the proposed algorithm based on the Glove tool, which is simple, efficient and has comparable performance to other word embedding models. Our approach is validated on several NLP tasks, and achieves promising results, especially in word similarity tasks.

2 Related Work

As a mature grammar applied in many NLP tasks, the post-processing is empirically validated on a variety of lexical-level intrinsic tasks (word similarity, concept categorization, word analogy) and sentence-level tasks (semantic textural similarity and text classification). It has also been applied to multiple datasets and with a variety of representation methods and hyperparameter choices in multiple languages. In each case, the processed representations are consistently better than the original ones. In extant studies, word embedding post-processing has been also used to composite local context in distributed representation learning models. For example, Labutov and Lipson [2013] proposed a fast method for re-embedding words from a source embedding s to a target embedding t by performing unconstrained optimization of a convex objective. On the other hand, Lee *et al*. [2016] used the Kolmogorov–Smirnov test to filter the anomalous dimensions during the process of the Glove training word vector, thereby improving the word representation effectiveness of the Glove model. Subsequently, Yu *et al*. [2018] added emotional information pertaining to specific words to the already trained word vectors and applied them to the sentiment analysis task. Mu and Viswanath [2018] re-projected the word vector by removing the non-zero mean vector from the pre-trained word vector. More recently, Wang *et al*. [Wang *et al*., 2018; Collell *et al*., 2017] consid-

ered the visual information corresponding to the text vocabulary in the word representation to obtain the visual vector corresponding to each vocabulary through the mapping relationship between the vocabulary and the visual vector space. In addition, the authors spliced the visual vector into the trained word vector of the Glove model to improve the effectiveness of word representation. These methods offer some very creative ways to obtain word embedding and achieve high performance on word similarity benchmarks. However, the geometric relationships between word and context representations underlying the aforementioned approaches remain insufficiently studied. Thus, the aim of the present work is to investigate such relationships in mathematics, when the similarity between words obtained by distributed word representation is inconsistent with the value based on human judgment and experience. Moreover, we propose an improved approach for training word representations based on our findings.

Recognizing that current word representation methods cannot effectively represent the semantic word similarity, Hashimoto *et al*. [2016] showed that word embedding and manifold learning are both suitable for recovering an Euclidean metric using co-occurrence counts and high-dimensional features, respectively. The authors pointed out that manifold learning can be used to map words from high-dimensional space to low-dimensional space. They further noted that the obtained word vector should serve as the input to distributed word representation. In this work, we follow a methodology which adheres to this paradigm, but employ distributed word representation to train the word vector, which is used to learn a manifold to improve the results. When using manifold learning to represent word vectors, we do not modify the word vector dimension, but transform between two equally-dimensional coordinate systems. The motivation behind this strategy is inspired by Hasan and Curry [2017], who also discussed the geometry of word representations.

Hasan and Curry [2017] sampled an off-the-shelf word embedding to provide input to the manifold learning process, which leverages local word neighborhoods formed in the original embedding space, learns the manifold, and embeds it into a new Euclidean space. The resulting re-embedding space is a recovery of a Euclidean metric space that is empirically superior to the original word embedding when tested on word similarity tasks. However, Hasan and Curry [2017] used local linear embedding of manifold learning to represent the word vectors, thus ignoring the effect of matrix of unfilled rank of each local neighborhood on the word representation when the number of neighbors exceeds the number of input dimensions. In our model, this issue is overcome by considering word representations in a more general sense.

3 The Proposed Method

We use manifold learning to represent a word vector, which can be formalized as $M_i = g(X_i)$, where g is a function represented by manifold under the circumstances of using i lexical text to represent X_i. Figure 4 shows our approach.

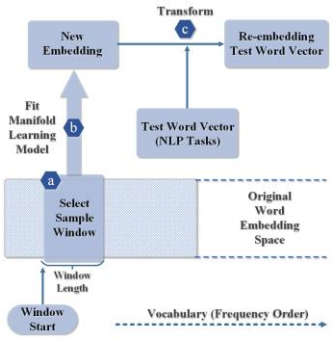

Figure 4: Refining Word Representations by Manifold Learning.

In Figure 4, we show the specific process by using manifold learning to re-embed word vectors trained for Glove. We start from an original embedding space with vectors ordered by words frequencies. In step (a), we select a subset of samples from the Glove model to train the manifold learning algorithm. In step (b), we train the manifold learning algorithm with the subset of samples selected in step (a), while retaining the dimensionality. In step (c), we correspond the vocabulary contained in the specific task to the word vector obtained in the Glove model, and re-embed the vocabulary by using the trained MLLE algorithm in step (b).

Discussion

In step (a), a sample subset of the words ordered by word frequencies is used. The rationale behind this approach is that word embedding attempts to recover a metric space, and frequent word co-occurrences can represent a better sampling of the underlying space due to their frequent usage, rather than being treated equally with other points. Thus, the manifold shape can be recovered more successfully. The sampling used here follows a sliding sample window to study the effect of its start position and size.

In step (c), we re-represent the word vector for the word vector sets obtained from the Glove model for a particular task. The process of re-representing the word vector is achieved by the manifold learning model trained in step (b) and the dimension of the word vector remain unchanged. This approach is adopted because, if all the word vectors trained in the Glove model are re-represented by manifold learning, the computational complexity will be high.

In order to better understand the relationship between word and context representations, we ensure that the similarity between the words in the Euclidean metric space is consistent with the semantic space. In this work, the Modified Locally Linear Embedding is adopted to re-embed the word representation. The motivation of our work is similar to that of the literature[Mu and Viswanath,2018], who also discussed the geometry of word representations. However, Hasan and Curry[2017] use the manifold learning algorithm LLE[Roweis and Saul, 2000] to represent the word vector without considering the problem that the matrix of unfilled rank in each local neighborhood when the number of neighbors exceeds the number of input dimensions. To solve

this problem, the local linear embedding algorithm incorporates an arbitrary regularization parameter r, the value of r is affected by the trace of the local weight matrix. Although it can be argued that $r \to 0$, indicating that the solution converges to the embedding case, there is no guarantee that $r > 0$ will hold under the optimal solution. This problem distorts the internal geometry of the manifold when embedded. In order to address this issue, the neighborhood adopts multiple weight vectors, for which Zhang and Wang [2006] propose the MLLE algorithm. For the purpose of obtaining a better word representation effect, in this work, the improved MLLE algorithm is applied to the Glove model.

We use the word vector contained in the sliding window selected in step (a) to train the improved MLLE algorithm. For a given word vector set $X = \{x_1, x_2, ..., x_N\}$, where N is the number of word vectors $X \in R^{m \times N}$ in the vocabulary, m is the dimension of the word vector. We use the K nearest neighbors to construct the neighbor structure of a word vector. MLLE algorithm constructs the word vector X and then represents the objective function as:

$$\min \left\| x_i - \sum_{j \in J_i} w_{j,i} x_j \right\|^2, \quad s.t. \sum_{j \in J_i} w_{j,i} = 1 \tag{1}$$

In the function given in Eq. (1), s_i is the number of approximation optimal weight vectors, $w_{j,i}^{\ell}$ is an x-adjacent weight vector. We use the geodesic distance to calculate the neighbors of each word vector. The specific formula is as follows:

$$d_{ij} = \frac{f(x_i, x_j)}{\sqrt{d(x_i) \cdot d(x_j)}} \tag{2}$$

where $f(x_i, x_j)$ is the geodesic distance between x_i and x_j, we use the dijkstra algorithm to calculate the geodesic distance between two points. $d(x_i)$, $d(x_j)$ are the mean distances of x_i and x_j from other points respectively.

To formulate the weight vector w_i consisting of the local weights $w_{j,i}$, $j \in J_i$, we rewrite Formula (1) and define it as follows:

$$G_i = [..., x_j - x_i] \tag{3}$$

$$x_i - \sum_{j \in J_i} w_{j,i} x_j = G_i w_i \tag{4}$$

$$\min \left\| G_i w_i \right\|^2, \quad s.t. \ w_i^T \mathbf{1}_{k_i} = 1 \tag{5}$$

For formula (5), Zhang and Wang [2006] apply svd to generate $k_i - r_i$ linearly independent weight vector $w_i^1, ..., w_i^{(k_i - r_i)}$. Where k_i is the number of neighborhood x_i, r_i is the regularization parameter($r_i > 0$).These weights are then used to construct a new embedding Y of the sample X via a neighborhood-preserving mapping by minimizing the cost function:

$$E(Y) = \min \sum_{i=1}^{N} \sum_{\ell=1}^{k_i - r_i} \left\| y_i - \sum_{j \in J_i} w_{j,i}^{\ell} y_j \right\|^2 \tag{6}$$

Algorithm 1 Refining Word Representations by Manifold Learning

Input:

1: Select a window in all word vectors as the data sample for manifold learning.

2: The data samples obtained in Step 1 are used to train the MLLE algorithm according to Eq. (1) and (6).
$X = \{x_1, x_2, \cdots, x_N\} \xrightarrow{fit} MLLE$.

3: The trained MLLE model is applied to test the words by re-embedding them according to Eq. (7) and (8) for $x(w_{test}) \to y(w_{test})$ (the word vector dimensions remain unchanged).

Output: Processed representations $y(w_{test})$.

In step (c), we use the model trained by Eq. (1) and (6) to re-embed the word vector x obtained from the Glove model for a specific task. The specific formula is as follows:

$$\min \left\| x - \sum_{j \in J_i} w_{j,i} x_j \right\|^2 \tag{7}$$

In Eq. (7), we apply svd to generate $k - r$ linearly independent weight vector $w^1, ..., w^{(k-r)}$. Where k is the number of neighborhood x, r is the regularization parameter($r > 0$). Then, we obtain the dense word embedding for x according to its local multiple and neighborhood:

$$E(y) = \min \sum_{\ell=1}^{k-r} \left\| y - \sum_{j \in J_i} w_{j,i}^{\ell} y_j \right\|^2 \tag{8}$$

Eq. (8) is solved to obtain the optimal y, which is the re-embedding result of the word vector x.

The word embedding algorithm based on manifold learning comprises of the steps given in Algorithm 1.

4 Experimental Setup

In order to verify the effectiveness of the model proposed in this paper, we conduct experiments on specific tasks related to several natural language processing, and the experimental setup and findings are discussed in the sections that follow.

4.1 Word Embedding

We use word vectors trained by the Glove model† as the original input, along with three corpora—the Common Crawl corpus consisting of 840B tokens and a vocabulary with 2.2M words (300-dimensional), Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, 50d, 100d, 200d, & 300d vectors), and Common Crawl (42B tokens, 1.9M vocab, 300d vectors)—in line with the approach adopted by Pennington *et al*. [2014b]. We train word vectors on these three corpora, respectively. This embedding choice is motivated by its state-of-the-art performance, which provides a strong base to learn the mappings.

†http://nlp.stanford.edu/projects/glove

4.2 Baseline Methods

The following word embedding models serve as the benchmark.

Glove. The Glove model [Pennington *et al*., 2014a] used the word co-occurrence matrix and takes into account the local and global features of words.

Lazaridou. Lazaridou *et al*. [2015] used a multimodal approach to add visual vectors to the original trained word vector.

Hasan. Hasan and Curry[2017] used the LLE algorithm to re-embed word vectors trained by Glove.

Collell. Collell *et al*.[2017] combines the text vector and visual vector of vocabulary to obtain the multi-modal vector representation of vocabulary.

Mu. This model [Mu and Viswanath, 2018] removes the non-zero mean vector from the pre-trained word vector and re-projection word vectors.

Wang. This method [Wang *et al*., 2018] assigns the visual vector weights when splicing the visual vector into the word vector.

4.3 Evaluation Tasks

We experiment with the method proposed in this paper and the baseline of Section 4.2 on two tasks, namely semantic relatedness and semantic similarity. Semantics-related tasks include the MEN dataset[Bruni *et al*., 2014], where 3,000 pairs of words are rated by crowd sourced participants, Wordrel-252 (WORDREL) [Agirre *et al*., 2009]; the MTurk dataset [Radinsky *et al*. 2011] where the 287 pairs of words are rated in terms of relatedness; Semantic similar task, as the first published RG65 dataset [Rubenstein *et al*., 1965]; the widely used WordSim-353 (WS353) dataset [Finkelstein *et al*., 2001] which contains 353 pairs of commonly used verbs and nouns; the SimLex-999 (SIMLEX) dataset [Hill and Korhonen, 2015]where the score measures "genuine" similarity; and the SimVerb-3500 (SIMVERB) dataset [Gerz *et al*., 2016], Wordsim-203 (WS203) [Gerz *et al*., 2016].

4.4 Evaluation Metrics

We use Spearman's method to evaluate the word representation of different models. The method calculates the Spearman rank correlation coefficient between the scorer's mark on the word pairs and the score of model acquisition representation:

$$\cos(u_1, u_2) = \frac{u_1 \cdot u_2}{\|u_1\| \cdot \|u_2\|} \tag{9}$$

$$r = p_{x,y} = \frac{\mathrm{cov}(x, y)}{\sigma_x \cdot \sigma_y} \tag{10}$$

In Eq. (9), cosine distance is used to measure the similarity of two words, where u_1 and u_2 represent two word vectors, respectively. Eq. (10) represents the Spearman rank correlation coefficient between the scorer's mark on the word pairs and the score of model acquisition representation. Here, the $\mathrm{cov}(x, y)$ represents the covariance between the ranked list x and y , and σ_x and σ_y represent the corresponding standard deviations, respectively. The more consistent the scoring of the model is with the scoring based on manual labeling, the higher the score is.

4.5 Model Settings

We use the scikit-learn toolkit for the experiments. First, we apply the Glove model to obtain the vector form of words contained in the natural language processing specific task, and then use the MLLE to re-represent the word vector. We do not update the entire vocabulary in this process because it is too computationally expensive. The test word does not contain all the vectors in the word list. When using MLLE to construct the neighborhood structure of the test words, we select a certain number of words from the vocabulary of the Glove model as the test training set. The size of the training word window was set as [1001, 1501, 2001]. The value range of the MLLE algorithm neighborhood is [300, 1000]. All models are trained in triplicate and the average results are reported in Table 1 and Table2.

Space	task	Pennington *et al*.,2014a	Hasan and Curry, 2017	Ours
6b50	WS353	61.2	56.6	**63.2**
6b50	RG65	60.2	53.0	**64.4**
6b100	WS353	64.5	64.3	**64.6**
6b100	RG65	65.3	67.3	**68.8**
6b200	WS353	68.5	**69.7**	67.0
6b200	RG65	75.5	76.0	**79.4**
6b300	WS353	65.8	**70.3**	67.9
6b300	RG65	75.5	80.5	**81.1**
42b300	WS353	75.2	78.4	**78.6**
42b300	RG65	80.0	83.4	**83.5**

Table 1: Spearman correlations (x100) between model predictions and human ratings on two evaluation datasets. Bold values represent the best result for each row of data. (window start ∈ [2000, 19001], number of MLLE local neighbors ∈ [1001, 2001], window length ∈ [300, 1001], manifold dimensionality = space dimensionality).

	RG65	WS353	MEN	SIMLEX	SIMVERB	MTURK	WS203	WORDREL
Pennington *et al*., 2014a	76.90	71.25	80.49	40.83	28.33	69.29	80.15	64.43
Lazaridou *et al*., 2015	75.00	--	--	40.00	--	--	--	--
Hasan and Curry, 2017	74.71	77.14	83.37	48.14	36.55	71.92	81.40	72.90
Collell *et al*., 2017	--	69.40	81.30	41.00	28.60	--	78.10	62.90
Mu and Viswanath, 2018	74.36	76.79	81.78	44.97	32.23	70.85	--	--
Wang *et al*., 2018	--	--	83.60	49.30	36.40	--	82.10	72.90
Ours	**77.19**	**78.40**	**84.19**	**49.40**	**37.32**	**72.78**	**82.32**	**73.69**
#inst	65	353	3000	999	3500	287	203	252

Table 2: Spearman correlations (x100) between model predictions and human ratings on eight evaluation datasets. Bold values denote the best result for each row of data and #inst is the number of words in each data set. (window start $\in$ [2000, 19001], number of MLLE local neighbors $\in$ [1001, 2001], window length $\in$ [300, 1001], manifold dimensionality = space dimensionality).

5 Results and Discussion

As shown in Table 1, we use the Glove model, the method proposed by Hasan and Curry[2017], and the method proposed in this paper, to conduct experiments on the data sets WS353 and RG65. The Glove model trains the word vectors with different dimensions in different corpora. Experiments are carried out on the obtained word vectors. The experimental results reported in Table 1 confirm that our proposed model clearly outperforms the baseline models. Experimental results relate to the method proposed by Hasan and Curry[2017]. The algorithm presented in this paper are superior to those of the Glove model in most cases, which also verifies the validity of manifold learning in word representation. As can be seen from the experimental results on data sets WS353 and RG65 with the corpus size of 6b and the word vector dimension of 50, the experimental results of the model proposed by Hasan and Curry[2017] are inferior to those obtained by the Glove model. However, the experimental results of the proposed method are significantly better than those of the Glove model and the model proposed by Hasan and Curry[2017]. When the word vector dimension is increased to 300, the models are ranked according to the experimental results on the data set WS353 with the corpus size of 6b. The model proposed by Hasan and Curry[2017]. has the best performance in this case, followed by the method presented in this work. In summary, the algorithm introduced in this paper can effectively improve the representation effect of the Glove model training word vector and also outperforms that proposed by Hasan and Curry[2017]. In some cases. These benefits are due to introducing manifold learning into the Glove model, while the proposed algorithm also mitigates the disadvantages of manifold learning in word representation. Therefore, the algorithm introduced in this paper has good generalization ability in word representation.

We conduct experiments on eight data sets with the Glove, multi-modal method and the method developed by Mu and Viswanath[2018] and the results are reported in Table 2. It can be seen that the experimental results of our proposed algorithm are better than those yielded by other algorithms. The models developed by Lazaridou *et al*. [2015], Collell *et al*.[2017]. and Wang *et al*.[2018]. are multimodal methods. These algorithms not only consider the word vector, but also splice the visual vector corresponding into the word vector when training the word vector. These three methods are thus better than the Glove model in terms of the experimental results. The algorithm proposed by Wang *et al*.[2018]. gives the visual vector weights while splicing the visual vector into the word vector. Therefore, from Table 2, it is obvious that the experimental results of the Wang *et al*. method are better than those of the Lazaridou *et al*.'s and Collell *et al*.'s models. Mu and Viswanath's method re-projects the word vector by removing the non-zero mean vector from the pre-trained word vector, which is better than the Glove model, according to the experimental results in Table 2.

6 Conclusions

In this paper, we introduced a simple, and yet counterintuitive post-processing technique. Distributed words representation suffers from inaccurate semantic similarity in the Euclidean metric space. Our technique uses the manifold learning to solve this problem, and thus renders off-the-shelf representations even stronger. The proposed algorithm is validated on eight datasets pertaining to semantic relativity and semantic similarity tasks Our method outperforms several state-of-the-art methods. Such a simple process could be used for word embedding in downstream tasks or as an initialization for training task-specific embedding. In the future, we will use manifolds to learn other algorithms in order to improve the representation of word vectors. We are also interested in investigating methods for utility exploiting for manifold learning word embedding for certain languages other than English (such as Chinese).

Acknowledgments

This work is partially supported by a grant from the Natural Science Foundation of China (No.61632011, 61572102,61702080, and 61602079) and the Fundamental Research Funds for the Central Universities (No. DUT18ZD102, No. DUT19RC(4)016) the National Key Research and Development Program of China (No. 2018YFC0832101), Postdoctoral Science Foundation of China (2018M631788).

References

[Agirre *et al*., 2009] Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, Aitor Soroa. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. *In Proc. of HLT-NAACL*, pages 19-27,2009.

[Bruni *et al*., 2014] Elia Bruni, Nam Khanh Tran, Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1-47,2014.

[Collobert *et al*., 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537,2011.

[Collobert and Weston, 2008] Ronan Collobert, Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. *In Proc. of ICML*, pages 160-167, 2008.

[Collell *et al*., 2017] Guillem Collell, Ted Zhang, Marie-Francine Moens. Imagined Visual Representations as Multimodal Embeddings. *In Proc. of AAAI*, pages 4378-4384, 2017.

[Finkelstein *et al*., 2001] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias. Placing search in context: The concept revisited. *In Proc. of WWW*, pages 406-414, 2001.

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

[Hasan and Curry, 2017] Souleiman Hasan, Edward Curry. Word Re-Embedding via Manifold Dimensionality Retention, *In Proc. of EMNLP*, pages 321–326, 2017.

[Hashimoto *et al*., 2016] Tatsunori B Hashimoto, David Alvarez-Melis, Tommi S Jaakkola. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, 4:273–286, 2016.

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

[Kira *et al*., 2011] Radinsky Kira, Eugene Agichtein, Evgeniy Gabrilovich. A word at a time: computing word relatedness using temporal semantic analysis. *In Proc. of WWW*, pages 337–346, 2011.

[Labutov and Lipson, 2016] lgor Labutov, Hod Lipson. Re-embedding words. *In Proc. of ACL*, pages 489–493, 2013.

[Lee *et al*., 2016] Yang-Yin Lee, Hao Ke, Hen-Hsen Huang, and Hsin-Hsi Chen. Less is More: Filtering Abnormal Dimensions in GloVe. *In Proc. of WWW*, pages 71-72, 2016.

[Lazaridou *et al*., 2015] Angeliki Lazaridou, Nghia The Pham, Marco Baroni. Combining language and vision with a multimodal skip-gram model. *In Proc. of NAACL*, pages 153–163,2015.

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

[Mikolov *et al*., 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient estimation of word representations in vector space. *In Proc. of ICLR*, 2013a.

[Mu and Viswanath, 2018] Jiaqi Mu, Suma Bhat, Pramod Viswanath. All-but-the-top: Simple and effective post-processing for word representations. *In Proc. of ICLR*, 2018.

[Pennington *et al*., 2014a] Jeffrey Pennington, Richard Socher, Christopher D. Manning. Glove: Global vectors for word representation. *In Proc. of EMNLP*, pages 1532–1543,2014a.

[Pennington *et al*., 2014b] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove resources. Available :http://nlp.stanford.edu/projects/glove/,2014b.

[Qiu *et al*., 2014] Lin Qiu, Yong Cao, Zaiqing Nie, Yong Yu, Yong Rui. Learning word representation considering proximity and ambiguity. *In Proc. of AAAI*, pages 1572–1578, 2014.

[Roweis and Saul, 2000] Sam T. Roweis, Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500): 2323-2326,2000.

[Rubenstein *et al*., 2000] Herbert Rubenstein, John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633,1965.

[Silberer and Lapata, 2014] Carina Silberer, Mirella Lapata. Learning grounded meaning representations with autoencoders. *In Proc. of ACL*, pages 721–732,2014.

[Wang *et al*., 2018] Shaonan Wang, Jiajun Zhang, Chengqing Zong. Learning Multimodal Word Representation via Dynamic Fusion Methods. *In Proc. of AAAI*, 2018.

[Yu *et al*., 2018] Liang-Chih Yu, Jin Wang, K. Robert Lai, Xuejie Zhang . Refining Word Embeddings Using Intensity Scores for Sentiment Analysis. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 26(3): 671-681,2018.

[Zhang and Wang, 2006] Zhenyue Zhang, Jing Wang. MLLE: Modified Locally Linear Embedding Using Multiple Weights. *In Proc. of NIPS*, pages 1593-1600,2006.