



Utilizing Local Tangent Information for Word Re-embedding

Wenyu Zhao^{1,3}, Dong Zhou¹ (✉) , Lin Li², and Jinjun Chen³

¹ School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, Hunan, China

wenyuzhao1993@hotmail.com, dongzhou1979@hotmail.com

² School of Computer Science and Technology,

Wuhan University of Technology, Wuhan 430070, Hubei, China

cathylilin@whut.edu.cn

³ Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Melbourne, VIC 3122, Australia

jinjun.chen@gmail.com

Abstract. Word embedding models typically learn dense and fixed-length vectors based on local word collocation patterns in a text corpus. Recent studies have discovered that these models often underestimate similarities between similar words and overestimate similarities between distant words. This leads to word similarity results obtained from word embedding models inconsistent with human judgment. A number of manifold learning-based word re-embedding methods are proposed to address this problem by re-embedding word vectors from the original embedding space to a new embedding space. However, these methods perform a weighted locally linear combination of embeddings of words and their neighbors twice. Besides, the reconstruction weights are easily influenced by the selection of word neighbors and the whole combination process is very time-consuming. In this paper, we introduce a novel word re-embedding method based on local tangent information to re-embed word vectors into a refined new space. Unlike previous approaches, our method re-embeds word vectors by aligning original and new embedding spaces based on the tangent information instead of performing weighted locally linear combination twice. To validate the proposed method, experiments were conducted on two standard evaluation tasks. The experimental results show that our method achieves better performance than state-of-the-art methods for word re-embedding.

Keywords: Word re-embedding · Local tangent information · Manifold learning

1 Introduction

Word embedding models represent words as dense and fixed-length vectors by mapping them from high-dimensional space to low-dimensional space. As the common knowledge, the distance between these dense vectors reflects the semantic relatedness of their

corresponding words. Furthermore, vectors generated by these models contain semantic and syntactic features, which are beneficial to mine the semantic relationships of words. Due to the ability of vector-space representations, word embedding models play an important role in a lot of Information Retrieval (IR) and Natural Language Processing (NLP) tasks, such as question answering [1], ad-hoc retrieval [2] and machine translation [3], part-of-speech tagging [4], named entity recognition [5], text classification [6]. Obviously, the discovery of semantic information is closely linked to the quality of word vectors. The representation quality of word vectors can directly affect the performance of a large amount of IR and NLP tasks as well.

Recently, a variety of word embedding models has been proposed to generate word embeddings, such as BERT [7], C&W [8], Continuous Bag-of-Words (CBOW) [9], Skip-Gram [9], GloVe [10] and other variants [11, 12]. BERT [7] and its variants [13, 14] can effectively produce contextual word embeddings with better support for different IR and NLP tasks. However, the computational cost is very high due to the huge amount of parameters. The refinement of contextual word embeddings will be studied in the future. In comparison with contextual models, static word embedding models are generally simple and efficient with a much lower computational cost. Although these static word embedding models can easily learn word vectors with linear structure data distribution, they fail to estimate similarities between words when the data distribution of words shows strong non-linear characteristics. They may underestimate similarities between nearby words and overestimate similarities between distant words, causing the problem about word similarity results obtained by word embedding models inconsistent with human judgment [15, 16].

As an example given in previous studies [15, 16], an example of the ground truth similarities between words obtained by human experience in a typical semantic similarity task is shown in Fig. 1. Another example of cosine similarity results of the same word pairs obtained by GloVe is shown in Fig. 2. As shown in these two Figures, the similarity result between “physics” and “proton” is more similar than that of “shore” and “woodland” based on human experience in Fig. 1. However, it achieves the opposite result in Fig. 2. The phenomenon fully reflects that similarity results between word pairs obtained by word embedding models may be inconsistent with human judgment.

$$\begin{aligned} \text{sim}(\text{"shore"}, \text{"woodland"}) &= 3.08 \\ &< \text{sim}(\text{"physics"}, \text{"proton"}) = 8.12 \end{aligned}$$

Fig. 1. Standard word similarity results judged by human beings

$$\begin{aligned} \text{sim}(\text{"shore"}, \text{"woodland"}) &= 0.36 \\ &> \text{sim}(\text{"physics"}, \text{"proton"}) = 0.33 \end{aligned}$$

Fig. 2. Word similarity results obtained by GloVe word embedding models

To address the similarity inconsistency problem, the existing studies show that re-embedding can rectify this problem by using manifold learning-based methods [15, 16]. Several approaches were proposed to re-embed word vectors into a new embedding space by using manifold learning-based methods for this purpose. For example, Locally Linear Embedding (LLE) [15] and Modified Locally Linear Embedding (MLLE) algorithms [16] were proposed to re-embed pre-trained GloVe word vectors into a new

embedding space. The above two methods both consider the local geometric information between words and their local neighboring words. They re-embed word vectors based on the weighted locally linear combination of words and their neighbors in both original and refined semantic spaces. Although they achieve good performance on word re-embedding, there exist certain demerits in both methods. On the one hand, the reconstruction weights can be easily affected by various options of word neighbors because these weights are generated by a linear combination of nearby words. On the other hand, these two methods need to perform the weighted locally linear combination twice in both two embedding spaces, which is time-consuming with high computation cost.

Unlike LLE and MLE methods, in this paper, we introduce a novel word re-embedding method based on Local Tangent Information (denoted as **LTI**) to re-embed word vectors into a refined new space. Our method firstly applies Principal Components Analysis (PCA) on word neighbors to construct a locally linear plane, which can be regarded as an approximation of the tangent information of these local words [17, 18]. Our **LTI** method then re-embeds word vectors by aligning original and refined new embedding space based on the local tangent information (containing different local geometric information). The proposed method can be more effective and efficient by directly aligning two embedding spaces based on local tangent information in comparison with LLE and MLE methods, which perform combination operation twice. To verify the proposed **LTI** method, we conduct several experiments on standard semantic relatedness and semantic similarity tasks. The experimental results show that our method achieves better performance than the state-of-the-art baseline methods for word re-embedding.

The contributions of our work are summarized as follows:

- We introduce a novel word re-embedding method based on local tangent information. Our method re-embeds word vectors by aligning original and refined semantic spaces based on the tangent information of words, which contains more geometric information and directly captures the relationships between original and refined embedding spaces.
- We are the first to demonstrate that local tangent information can be used to improve the performance of word re-embedding.
- We conduct several experiments to validate our proposed method in this paper. Compared with the state-of-the-art baseline methods of word re-embedding, the results show that our proposed method can achieve better performance by utilizing local tangent information of words and their neighbors.

The rest of our paper is organized as follows: Sect. 2 describes the related work. Our method is presented in Sect. 3. Section 4 shows the details of experimental settings. In Sect. 5, we provide and analyze the experimental results. Finally, Sect. 6 concludes the paper and discusses future research.

2 Related Work

2.1 Count-Based Word Embedding Methods

Count-based word embedding methods only focus on word co-occurrence probability or word counts. Vector space model is the early idea to use vectors to express words [19].

This method constructed a word-document co-occurrence matrix and used it to represent words and documents as vectors by using TF-IDF. However, this method does not consider the true semantic information of words. Latent Semantic Analysis (LSA) [20] can also generate word embeddings by applying Singular Value Decomposition (SVD) to a word-document matrix. Subsequently, Lund and Burgess [21] proposed a Hyperspace Analogue to Language (HAL) model that constructed a word-context word matrix based on a corpus to form vector representations. Dhillon et al. [22] introduced an alternative method leveraging Canonical Correlation Analysis (CCA) between left and right contexts to generate word embeddings. Lebrete and Collobert [23] used Hellinger PCA to the word-context matrix to obtain word embeddings. In summary, these methods globally utilize word-context co-occurrence or counts to produce word embeddings based on word-context matrices in a corpus. Though the aforementioned methods are simple and effective, these count-based methods only consider the co-occurrence probability or word counts between words and their context words rather than the real semantic relationships between them.

2.2 Prediction-Based Word Embedding Methods

Prediction-based word embedding methods generate word embeddings by using the contexts of words. In the early time, Hinton proposed a word distributed representation hypothesis [24]. Most of the subsequent methods are inspired by this hypothesis. They represent words as distributional dense, fixed-length and low-dimensional word vectors. Bengio et al. [25] proposed an N-Gram neural network language model and used it to generate word embeddings. In this method, embeddings are a by-product during training a neural network language model (NNLM). Bengio and Senecal [26] improved NNLM by using a Menote Carlo method and hierarchical softmax layer to speed up word embedding generation. Similarly, Mnih and Hinton [27] proposed a slightly modified log-bilinear model to produce word embeddings. As word embeddings are by-products of previous models, Collobert and Weston [28] designed a model solely aimed at generating word embeddings by using unlabeled data. Following these mentioned works, Collobert et al. proposed a unified neural network architecture C&W and a learning algorithm to discover internal representations of words [8]. Mikolov et al. presented two famous model architectures for learning high-quality continuous vector representations for words [9]. One model (CBOW) predicts the current word by utilizing the context of this word. Another model (Skip-gram) predicts the surrounding words based on the current word. Inspired by Skip-gram and CBOW, Qiu et al. proposed two variants of the CBOW model and the Skip-gram model to produce high-quality distributed representations for words by considering both word proximity and ambiguity [11]. Similar to these studies, Pennington et al. [10] proposed a GloVe model that combines the global features of a corpus and the local contextual features of words for generating word representations.

Apart from the static word embedding models described above, several contextual embedding models have been proven to be effective for word embedding generation these days, such as BERT [7] and its variants [13, 14]. Though word embeddings generated from such models can provide good support for different IR and NLP tasks, the computational cost is very high due to the huge amount of parameters. On the contrary,

static word embedding models are simpler and more efficient with a much lower computational cost. In this paper, we mainly focus on static word embeddings and leave the study of refining contextual word embeddings as the future work.

2.3 Word Vector Re-embedding Methods

Many studies are focusing on re-embedding word vectors for improving the quality of word vectors. For example, Chaudhary et al. adapted continuous word representations by using morphological and phonological subword representations for low-resourced languages [29]. Kolyvakis et al. utilized a novel entity alignment method called DeepAlignment to refine pre-trained word vectors for generating ontological entity descriptions in the ontology matching task [30]. Seyeditabari et al. incorporated emotional information of words into pre-trained word vectors for generating emotional embeddings, which can capture the emotional contents of words [31]. Utsumi proposed a simple method to re-embed pre-trained word embeddings by using layer-wise relevance propagation [32]. Yu et al. presented an improved word vector model to refine existing pre-trained word vectors by leveraging real-valued sentiment intensity scores provided by sentiment lexicons [33].

However, this paper mainly focuses on studies about word vector re-embedding by re-mapping word vectors from the original embedding space to a new refined embedding space. Mu et al. projected word embeddings by removing the common mean vectors of pre-trained word vectors [34]. Some methods focus on exploring the geometric structure of word embeddings by using manifold-learning based algorithms and they show that reconstruction of word embeddings can capture the underlying manifold of the data [15, 16, 35]. Hasan and Curry utilized word neighbors in the original embedding space to re-embed pre-trained GloVe vectors into a new embedding space based on LLE [15]. The re-embedded word vectors could learn rich semantic information of word embeddings from a new embedding space for addressing the word similarity inconsistency issue. Furthermore, Chu et al. used a Modified Locally Linear Embedding (MLLE) algorithm to refine word representations in the aspect of geometric information of words and their neighbors [16].

Although the aforementioned manifold learning algorithms for word re-embedding have been proven to be effective, these methods need to perform the weighted locally linear combinations twice in both original and refined embedding spaces. Unlike these methods, we approach the problem of word re-embedding by utilizing local tangent information of words. This information can directly capture the relationships between the original and new embedding space instead of relying on local weights. Our method also avoids performing a locally linear combination of nearby words twice.

3 A Novel Word Re-embedding Method

3.1 Overall Framework

The overall framework of our proposed method based on Local Tangent Information (LTI) is shown in Fig. 3. There are four main steps in our method. In step (a), we choose

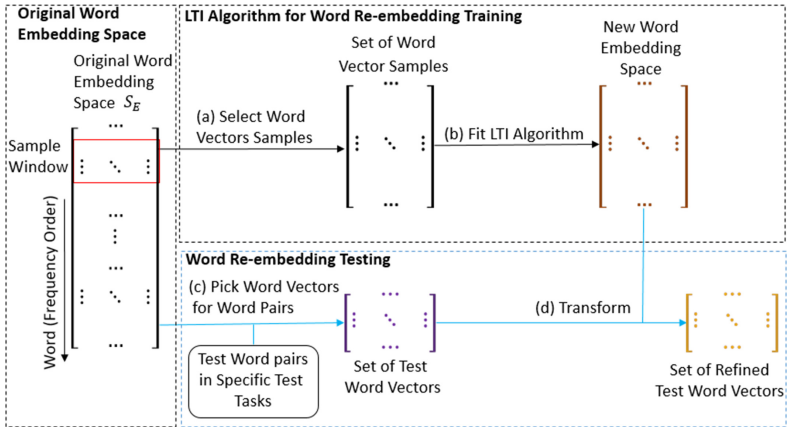


Fig. 3. The framework of our proposed method

a subset of word vector samples from the original embedding space by using a sample window. Word vectors are ordered according to their correspondent word frequencies (frequent word co-occurrences) in this corpus. Note that as in previous studies [15, 16], ordering word vectors and selecting samples instead of using all vectors can avoid a high computational cost. In our work, the original embedding space we used is trained by GloVe, because the pre-trained word vectors from this model can effectively represent words by considering contextual features of words and global features of a corpus in comparison with other static word embedding generation models. In step (b), we train a Local Tangent Information algorithm (**LTI**) on these selected samples in step (a) and this fitted manifold learning algorithm will be used to transform word vectors from original embedding space to a new refined embedding space. In this process, we just transform between two equally-dimensional coordinate systems and keep the dimension of word vectors unchanged. In step (c), we obtain word vectors of test word pairs (test word pairs from specific tasks to validate the effect of word re-embedding) from the original embedding space. In step (d), we re-embed these test word vectors into a new re-embedding space to obtain new vectors by using the fitted **LTI** obtained in step (b).

3.2 Word Re-embedding Based on Local Tangent Information

LLE [15] and MLLLE [16] methods aim at addressing the problem that word similarity results of word pairs obtained by word embedding models are inconsistent with that determined by human beings through word re-embedding. These two methods re-embed word vectors by preserving local geometric information of words and their neighbors. However, their research has certain limitations that the reconstruction weights are easily influenced and these two methods need to perform the weighted locally linear combination twice in both two embedding spaces.

Algorithm 1. Word Re-embedding Algorithm based on LTI

Input: original word embedding space \mathcal{S} , test words $\{w_1, w_2, \dots, w_m\}$
Output: refined word representations set \mathcal{Z} of test words

- 1: choose word vector samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ from \mathcal{S}
- 2: **for** each $\mathbf{X} \in \mathcal{S}$ **do**
- 3: according to Eq. (1), (5) and (6), fit \mathbf{X} to obtain new word embedding space \mathcal{Y}
- 4: **end for**
- 5: **for** all $w \in \{w_1, w_2, \dots, w_m\}$ **do**
- 6: obtain word vectors of w from \mathcal{S}
- 7: re-embed vector of w to obtain refined vector set \mathcal{Z} based on \mathcal{Y}
- 8: **end for**
- 9: return refined word representations set \mathcal{Z} of test words

Unlike LLE and MLLE methods, our proposed method uses local geometric information different from those of the above two methods. To address the limitations brought by their methods, in this paper, we introduce a novel word re-embedding method based on Local Tangent Information (denoted as **LTI**) to re-embed word vectors into a refined new space. To be specific, a locally linear plane is constructed by leveraging PCA on word neighbors. It is considered as an approximation of the tangent information at each word point [17, 18]. Since both the original and new embedding spaces exist a linear mapping of each word from their spaces to the local tangent information, our method aligns these linear mappings based on local tangent information to re-embed word representations.

As we mentioned in the last subsection, word vector samples are firstly chosen from pre-trained GloVe word vector corpus (original embedding space \mathcal{S}) through a simple window and Local Tangent Information (**LTI**) is trained on these samples. The set of selected samples is defined as a word vector set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, where $\mathbf{X} \in R^{d \times N}$, N is the number of words and d represents the dimension of word vectors. In our proposed method, for each word vector \mathbf{x}_i , ($i = 1, 2, \dots, N$), we find its k nearest neighborhoods (including \mathbf{x}_i itself) and denote the adjacent neighborhood set as $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ik}]$. Subsequently, for each word vector \mathbf{x}_i , we apply PCA to each neighborhood set \mathbf{X}_i to approximate the local tangent information of the word corresponding to a word vector \mathbf{x}_i for preserving the local structure of the neighborhood set \mathbf{X}_i of \mathbf{x}_i . The objective function is

$$\arg \min_{\mathbf{Q}_i, \theta_i} \sum_{j=1}^k \|(\mathbf{x}_{ij} - \mathbf{x}) - \mathbf{Q}\theta_{ij}\|^2 = \arg \min_{\mathbf{Q}_i, \boldsymbol{\Omega}_i} \|\mathbf{X}_i \mathbf{H}_k - \mathbf{Q}\boldsymbol{\Omega}_i\|^2 \quad (1)$$

where $\mathbf{H}_k = \mathbf{I} - \frac{\mathbf{e}\mathbf{e}^T}{k}$ is centralization matrix, \mathbf{I} is an identity matrix, \mathbf{e} means the vector of all 1's, \mathbf{Q} is an orthonormal basis matrix of the tangent information, $\boldsymbol{\Omega}_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{ik}]$ represents a local linear approximation of \mathbf{X}_i , i.e. θ_{ij} is the tangent coordinate corresponding to the orthonormal basis matrix \mathbf{Q}_i . Apparently, the optimal \mathbf{x} is the mean value of all neighborhood words vectors \mathbf{x}_{ij} , ($j = 1, 2, \dots, k$) of the sample point \mathbf{x}_i , ($i = 1, 2, \dots, N$). The optimal \mathbf{Q} is given by \mathbf{Q}_i and it is made up of t left

singular vectors of $X_i H_k$ corresponding to its t largest singular values (t is equal to d , as the embedding dimension is the same in both two embedding spaces.) The tangent coordinates Ω_i can be computed as

$$\Omega_i = Q_i^T X_i H_k \quad (2)$$

After obtaining the local tangent coordinates, we have to construct the global coordinates in a new embedding space. The purpose of the global arrangement of local tangent information is to find a group of new space coordinates $Y = [y_1, y_2, \dots, y_N]$, which are called global coordinates in a new embedding space. Therefore, we assume that there is a projection matrix, which re-embeds tangent coordinates Ω_i to new space coordinates $Y_i = \{y_{i1}, y_{i2}, \dots, y_{iN}\}$, then we have

$$Y_i H_k = L_i \Omega_i + E_i \quad (3)$$

where L_i is the projection matrix mapping Ω_i to Y_i and E_i is the local reconstruction error term. To preserve as much of the local geometry in a new embedding space as possible, we intend to find Y_i and L_i by minimizing the reconstruction error E_i

$$\arg \min_Y \sum_{i=1}^N \|E_i\|^2 = \arg \min_Y \sum_{i=1}^N \|Y_i H_k - L_i \Omega_i\|^2 \quad (4)$$

Obviously, the mapping error is minimal when $L_i = Y_i H_k \Omega_i^+$, where Ω_i^+ is Moore-Penrose generalized inverse of Ω_i . Let refined word vector set $Y = [y_1, y_2, \dots, y_N]$ be the d dimensional global coordinates of all words in X (Y also be refined new embedding space) and ϕ_i be the 0-1 selection matrix such that $Y \phi_i = Y_i$. The optimal Y can be achieved by minimizing the overall reconstruction error of all neighborhoods and the Formula (4) can be rewritten as:

$$\begin{aligned} \arg \min_Y \sum_{i=1}^N \|E_i\|^2 &= \arg \min_Y \sum_{i=1}^N \|Y_i \phi_i W_i\|^2 \\ &= \text{mintrace}(Y \phi W W^T \phi^T Y^T) \\ &= \text{mintrace}(Y B Y^T) \end{aligned} \quad (5)$$

where $\phi = [\phi_1, \phi_2, \dots, \phi_N]$, $W = \text{diag}(W_1, W_2, \dots, W_N)$ with $W_i = H_k(I - \Omega_i^+ \Omega_i)$ and $B = \phi W W^T \phi^T$. In order to uniquely obtain Y , we will impose the constraint $Y Y^T = I$. The refined new word vector set Y is composed of the t eigenvectors of the matrix B , and these eigenvectors correspond to the 2nd to $(t+1)$ th smallest eigenvalues of B . Then the eigenvector matrix picked from B is $[u_2, \dots, u_{t+1}]$, where u_i is an eigenvector of B . Thus, d dimensional refined new embedding set Y should be:

$$Y = [u_2, \dots, u_{t+1}] \quad (6)$$

In our work, we firstly use word vectors samples from the original embedding space to train the **LTI** algorithm by Eq. (1), Eq. (5) and Eq. (6) to obtain a new embedding space Y . Then we can obtain the refined new embedding set of test word vectors in specific tasks by using the new embedding space Y . The overall procedure of our Word Re-embedding Algorithm based on **LTI** is described in Algorithm 1.

4 Experimental Setup

4.1 Data Description

As we mentioned before, we use the original word vectors trained by GloVe [10]. Moreover, we use two sets of GloVe word vectors¹. One is trained from Wikipedia 2014+Giga-word 5 (consists of 6 Billion tokens, 400,000 vocabularies, word vectors with 50, 100, 200, and 300 dimensions, denoted as 6B50/100/200/300d). Another set is trained from Common Crawl (consists of 42 Billion tokens, 1.9 Million vocabularies, word vectors with 300 dimensions, denoted as 42B300d). To demonstrate the effectiveness of our proposed method, we conduct experiments on semantic relatedness and semantic similarity tasks. The semantic relatedness task focuses on the degree of semantic relatedness between words. It contains three datasets, including MEN dataset (3000 word pairs) [36], WordRel (WordRel) dataset (252 word pairs) [37], MTurk (MTurk) dataset (287 word pairs) [38]. The semantic similarity task pays attention to the degree of semantic similarity between words. It includes four datasets, which are RG65 (RG) dataset (65 word pairs) [39], WordSim-353 (WS353) dataset (353 word pairs) [40], SimLex-999 (SimLex) dataset (999 word pairs) [41], and WordSim-203 (WS203) dataset (203 word pairs) [42] respectively.

4.2 Baselines

We validate our proposed method for word re-embedding by comparing it with the following representative baseline methods.

GloVe. It is the original GloVe method [10]. This distributed word representation method is general and quite effective. The word vectors trained by this method consider local features of contextual words and global features of a corpus.

LLE. Hasan and Curry utilized local word neighbors to re-embed pre-trained word vectors (also trained by GloVe) based on the LLE manifold learning algorithm [15].

RoM. Mu et al. removed the common mean vectors of the pre-trained word vectors and the top principal components of all words for post-processing word vectors [34].

MLLE. Similar to [15], Chu et al. used the MLLE manifold learning algorithm to re-embed word vectors trained by GloVe [16].

LTI. The method proposed in the current paper. We use a manifold learning method that utilizing local tangent information of words and their neighbors to re-embed word vectors by aligning the original and new embedding space based on the tangent information of words.

4.3 Evaluation Metrics

To evaluate the performance of our proposed method and baseline methods, we adopt Spearman's method to compute the Spearman Rank Correlation coefficient between word similarity scores (similarity scores of word pairs obtained from word re-embedding

¹ <http://nlp.stanford.edu/projects/glove>.

methods) with human judgments (original similarity scores of word pairs in each dataset). The Spearman Rank Correlation is defined as:

$$\cos(u_1, u_2) = \frac{u_1 \cdot u_2}{\|u_1\| \cdot \|u_2\|} \quad (7)$$

$$r = p_{x,y} = \frac{\text{cov}(x, y)}{\sigma_x \cdot \sigma_y} \quad (8)$$

Equation (7) is used to calculate the similarity results of each pair of words in specific tasks, where u_1 and u_2 represent two word vectors. Equation (8) represents the Spearman Rank Correlation coefficient between word similarity scores and human ratings, $\text{cov}(x, y)$ represents the covariance between the score ranking list x and y , which denote the score list of word similarity scores obtained by word re-embedding methods and the score list of human judgments respectively, σ_x and σ_y represent the corresponding standard deviations of these two score lists. The more consistent similarities of word pairs obtained by word re-embedding methods with human judgments, the higher the Spearman score is.

4.4 Implementation Details

Firstly, we select word vector samples from a pre-trained word vector corpus by using a sample window and use the **LTI** algorithm to train these samples. Then for each specific task, we obtain word vectors of test word pairs and transform these word vectors into a new embedding space by using the fitted **LTI** algorithm. Finally, we compute cosine similarity scores of word pairs in each specific task and compute the Spearman scores. In our method, the range value of number of neighbors chosen was set as [300, 1500] and the step is 100. The range value of the training sample window size was set as [300, 2000] and the step is 50. Previous experiments show that the best sample size should be as close as possible to the number of neighbors because a wider range has no significant difference in results and has high time and computation cost.

5 Results and Discussion

5.1 Performance on Word Vectors with Different Embedding Dimensions

In order to evaluate the performance of our proposed method and other word re-embedding methods on word vectors with different embedding dimensions, we conduct experiments on WS353 and RG dataset as in previous studies [15, 16]. The experimental results are shown in Table 1. As shown in this table, LLE, MLLE and our proposed **LTI** method perform better than GloVe in most cases. This demonstrates that using a manifold-learning based algorithm is beneficial to generate word embeddings with high quality. Furthermore, we can observe that our proposed method achieves better performance than LLE and MLLE methods in 5 out of 10 experimental runs. In terms of dataset, the MLLE method achieves good performance on RG dataset than that of WS353 dataset. We can observe that our proposed method achieves the best result in 4

out of 5 experimental runs on RG dataset. However, this proposed method only obtains the highest scores in 2 out of 5 experimental runs on WS353 dataset. This is probably due to some noises existing in word vectors in the due dataset. Another reason is that the distance of words and their neighbors in RG dataset may be closer than that of words and their neighbors in WS353 dataset, so the geometric information of RG dataset is more beneficial to the manifold-learning based methods for word re-embedding than that of WS353 dataset.

Table 1. Spearman correlations scores of various methods on two evaluation datasets. Bold values represent that our method achieves the best results than baseline methods. Note that baseline results are taken from [16].

Space	Task	GloVe	LLE	MLLE	LTI
6B50d	WS353	61.2	56.6	63.2	61.2
6B100d	WS353	64.5	64.3	64.6	66.4
6B200d	WS353	68.5	69.7	67.0	68.2
6B300d	WS353	65.8	70.3	67.9	69.3
42B300d	WS353	75.2	78.4	78.6	78.6
Space	Task	GloVe	LLE	MLLE	LTI
6B50d	RG	60.2	53.0	64.4	62.6
6B100d	RG	65.3	67.3	68.8	73.3
6B200d	RG	75.5	76.0	79.4	81.5
6B300d	RG	75.5	80.5	81.1	83.1
42B300d	RG	80.0	83.4	83.5	86.5

In addition, with regard to embedding dimensions, our proposed method outperforms the MLLE method on both datasets with embedding dimensions more than 50. However, when RG and WS353 datasets containing 6B tokens and the embedding dimension is 50, the MLLE shows better performance than our method. The reason may be that multiple weights are more suitable to describe the relationships between words and their neighbors than the tangent information when the embedding dimension is very low. However, when RG dataset contains 6B tokens and the embedding dimension increases, our proposed method shows better performance than all baseline methods. It is obvious that the higher dimension of word vectors is, the better performance of our proposed method can get because word vectors with high dimensions can capture more semantic information.

Moreover, we can observe that when the size of datasets increases (from 6B to 42B) and the embedding dimension reaches 300, our proposed method can greatly improve word similarity performance on both datasets. This indicates that the larger training size and larger dimension are beneficial for word re-embedding. Then, we conduct experiments on seven datasets with a size of 42B and use the embedding dimension of 300 to further validate the effectiveness of our proposed **LTI** method.

5.2 Performance on Two Evaluation Tasks

In addition to these experiments, more experiments are conducted on seven datasets to further validate the performance of our proposed **LTI** method. Table 2 displays the results of all methods in two evaluation tasks. From this table, an observation is that almost all word re-embedding methods (**LTI**, **MLLE**, **LLE** and **RoM**) perform better than GloVe. These results are in-line with previous findings so that these two tasks are quite suitable to evaluate the word re-embedding methods. This further suggests that word re-embedding can improve the performance of word representations.

Table 2. Spearman correlations between scores predicted by our method and scores obtained from human judgment on two evaluation tasks. Bold values represent that our method achieves the best results than baseline methods. Note that baseline results are taken from [16].

Method	Semantic similarity task				Semantic relatedness task		
	RG	WS353	SimLex	WS203	MTurk	WordRel	MEN
GloVe	76.90	71.25	40.83	80.15	69.29	64.43	80.49
LLE	74.71	77.14	48.14	81.40	71.92	72.90	83.37
RoM	74.36	76.79	44.97	–	70.85	–	81.78
MLLE	77.19	78.40	49.40	82.32	72.78	73.69	84.19
LTI	86.48	78.58	50.46	81.92	73.15	74.65	83.50

We notice that our proposed **LTI** method is the best performing method on 5 out of 7 datasets in comparison with the **MLLE** method. This is because the **MLLE** method may be strongly influenced by the local weights. Our method aligns the original and refined semantic space based on the local tangent information rather than the multiple local weights. Furthermore, our method does not calculate the weighted combination of embedding of words and their neighbors twice, which is more efficient. Our **LTI** method performs slightly worse than the **MLLE** method on **WS203** and **MEN** datasets. This is likely caused by the better effect of the local weights in the **MLLE** method. However, the differences are quite small (0.49% and 0.83%). In summary, our method can achieve better performance than all other baseline methods and it is more computationally efficient than all previously proposed word re-embedding methods that are included in the comparison.

6 Conclusion

Word re-embedding can address the problem that the similarity scores of word pairs obtained by word embedding models are inconsistent with human ratings. In this paper, we introduce a novel word re-embedding method based on Local Tangent Information (**LTI**) to re-embed word vectors. Our **LTI** method tries to re-embed vectors by aligning the original and new embedding spaces based on local tangent information.

We conduct several experiments on semantic relatedness and semantic similarity tasks. The results demonstrate that our proposed method achieves better performance than the existing word re-embedding methods. In future work, our method can be advanced in two directions. On the one hand, we will try to discover the key factors that influence the effectiveness of the word re-embedding process. On the other hand, we will explore the contextual word embedding refinement by using manifold learning methods.

Acknowledgements. We would like to thank anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China under Project No. 61876062 and General Key Laboratory for Complex System Simulation under Project No. XM2020XT1004.

References

1. Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., Fujita, H.: Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Inf. Sci.* **514**, 88–105 (2020)
2. Bagheri, E., Ensan, F., Al-Obeidat, F.: Neural word and entity embeddings for ad hoc retrieval. *Inf. Process. Manag.* **54**(4), 657–673 (2018)
3. Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., Makhoul, J.: Fast and robust neural network joint models for statistical machine translation. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1, pp. 1370–1380 (2014)
4. Collobert, R.: Deep learning for efficient discriminative parsing. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 224–232 (2011)
5. Turian, J., Lev, R., Yoshua, B.: Word representations: a simple and general method for semi-supervised learning. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 384–394. Association for Computational Linguistics (ACL) (2010)
6. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment tree-bank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, vol. 1631, pp. 1631–1642. Citeseer (2013b)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, pp. 4171–4186 (2018)
8. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**(1), 2493–2537 (2011)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, pp. 3111–3119 (2013a)
10. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)
11. Qiu, L., Cao, Y., Nie, Z., Yu, Y., Rui, Y.: Learning word representation considering proximity and ambiguity. In: *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1572–1578 (2014)

12. Peng, X., Zhou, D.: A framework for learning cross-lingual word embedding with topics. In: Wang, X., Zhang, R., Lee, Y.-K., Sun, L., Moon, Y.-S. (eds.) APWeb-WAIM 2020. LNCS, vol. 12318, pp. 285–293. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60290-1_22
13. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. In: Proceedings of the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, pp. 1–17 (2020)
14. Liu, W., Zhou, P., Wang, Z., Zhao, Z., Deng, H., Ju, Q.: FastBERT: a self-distilling BERT with adaptive inference time. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 6035–6044 (2020)
15. Hasan, S., Curry, E.: Word re-embedding via manifold dimensionality retention. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 321–326 (2017)
16. Chu, Y., Lin, H., Yang, L., Diao, Y., Zhang, S., Fan, X.: Refining word representations by manifold learning. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 5394–5400 (2019)
17. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *J. Shanghai Univ.* **8**(4), 406–424 (2002)
18. Zhang, Z., Zha, H.: Nonlinear dimension reduction via local tangent space alignment. In: Liu, J., Cheung, Y., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 477–481. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45080-1_66
19. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* **18**(11), 613–620 (1975)
20. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **41**(6), 391–407 (1990)
21. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical co-occurrence. *Behav. Res. Methods Instrum. Comput.* **28**(2), 203–208 (1996)
22. Dhillion, P., Foster, D.P., Ungar, L.H.: Multi-view learning of word embeddings via CCA. *Advances in Neural Information Processing Systems*, pp. 199–207 (2011)
23. Le Bret, R., Collobert, R.: Word embeddings through hellinger PCA. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. Idiap, pp. 482–490 (2013)
24. Hinton, G.E.: Learning distributed representations of concepts. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, vol. 1, pp. 1–12 (1986)
25. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**(1), 1137–1155 (2003)
26. Bengio, Y., Senécal, J.S.: Quick training of probabilistic neural nets by importance sampling. In: Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS), pp. 1–9 (2003)
27. Mnih, A., Hinton, G.: Three new graphical models for statistical language modelling. In: Proceedings of the 24th International Conference on Machine Learning, pp. 641–648 (2007)
28. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167 (2008)
29. Chaudhary, A., Zhou, C., Levin, L., Neubig, G., Mortensen, D.R., Carbonell, J.G.: Adapting word embeddings to new languages with morphological and phonological subword representations. arXiv preprint [arXiv:1808.09500](https://arxiv.org/abs/1808.09500) (2018)
30. Kolyvakis, P., Kalousis, A., Kiritsis, D.: Deepalignment: Unsupervised ontology matching with refined word vectors. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 787–798 (2018)

31. Seyeditabari, A., Tabari, N., Gholizadeh, S., Zadrozny, W.: Emotional embeddings: refining word embeddings to capture emotional content of words. arXiv preprint [arXiv:1906.00112](https://arxiv.org/abs/1906.00112) (2019)
32. Utsumi, A.: Refining pretrained word embeddings using layer-wise relevance propagation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4840–4846 (2018)
33. Yu, L.C., Wang, J., Lai, K.R., Zhang, X.: Refining word embeddings using intensity scores for sentiment analysis. *IEEE Trans. Audio Speech Lang. Process.* **26**(3), 671–681 (2017)
34. Mu, J., Bhat, S., Viswanath, P.: All-but-the-top: simple and effective postprocessing for word representations. In: Proceedings of Poster at 6th International Conference on Learning Representations (ICLR), 1–25 (2018)
35. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
36. Bruni, E., Tran, N.K., Baroni, M.: Multimodal distributional semantics. *J. Artif. Intell. Res.* **49**(1), 1–47 (2014)
37. Agirre, E., Alfonseca, E., Hall, K.B., Kravalova, J., Pasca, M., Soroa, A.: A study on similarity and relatedness using distributional and wordnet-based approaches. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pp. 19–27 (2009)
38. Kira, R., Agichtein, E., Gabrilovich, E.: A word at a time: computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th International Conference on World Wide Web (WWW), pp. 337–346 (2011)
39. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. *Commun. ACM* **8**(10), 627–633 (1965)
40. Finkelstein, L., Gabrilovich, E., Matias, Y.: Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web (WWW), pp. 406–414 (2001)
41. Hill, F., Korhonen, A.: Simlex-999: evaluating semantic models with (genuine) similarity estimation. *Comput. Linguist.* **41**(4), 665–695 (2015)
42. Gerz, D., Vulić, I., Hill, F., Reichart, R., Korhonen, A.: Simverb-3500: a large-scale evaluation set of verb similarity. arXiv preprint [arXiv:1608.00869](https://arxiv.org/abs/1608.00869) (2016)