Manifold Learning-based Word Representation Refinement Incorporating Global and Local Information

Wenyu Zhao [1, 3], Dong Zhou[1, *], Lin Li[2], Jinjun Chen[3]

1. School of Computer Science and Engineering,
Hunan University of Science and Technology, Xiangtan, Hunan, 411201, China
2. School of Computer Science and Technology,
Wuhan University of Technology, Wuhan, Hubei, 430070, China
3. Department of Computer Science and Software Engineering,
Swinburne University of Technology, Hawthorn, Melbourne, VIC, 3122, Australia

{wenyuzhao1993, dongzhou1979}@hotmail.com
cathylilin@whut.edu.cn
jinjun.chen@gmail.com

Abstract

Recent studies show that word embedding models often underestimate similarities between similar words and overestimate similarities between distant words. This results in word similarity results obtained from embedding models inconsistent with human judgment. Manifold learning-based methods are widely utilized to refine word representations by re-embedding word vectors from the original embedding space to a new refined semantic space. These methods mainly focus on preserving local geometry information through performing weighted locally linear combination between words and their neighbors twice. However, these reconstruction weights are easily influenced by different selections of neighboring words and the whole combination process is time-consuming. In this paper, we propose two novel word representation refinement methods leveraging isometry feature mapping and local tangent space respectively. Unlike previous methods, our first method corrects pre-trained word embeddings by preserving global geometry information of all words instead of local geometry information between words and their neighbors. Our second method refines word representations by aligning original and refined embedding spaces based on local tangent space instead of performing weighted locally linear combination twice. Experimental results obtained from standard semantic relatedness and semantic similarity tasks show that our methods outperform various state-of-the-art baselines for word representation refinement.

1 Introduction

Semantic word representations are normally represented as dense, distributed and fixed-length word vectors that are generated by different word embedding models. They can be used to discover some semantic information among words and measure the semantic relatedness of words. Not surprisingly, word vectors and word embedding models have been attracting a lot of attention in the research com-

* Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3401–3412
Barcelona, Spain (Online), December 8-13, 2020

munity. These word embeddings have been proved to be quite useful in a number of Information Retrieval (IR) tasks, such as machine translation (Mujjiga et al., 2019), text classification (Stein et al.,2019), question answering (Esposito et al., 2020) and ad-hoc retrieval (Bagheri et al., 2018; Roy et al. 2018).

The performance of aforementioned tasks critically depends on the quality of word embeddings generated from different models. There exist a large number of word embedding models such as BERT (Devlin et al., 2019), C&W (Collobert et al., 2011), Continuous Bag-of-Words (CBOW) (Mikolov et al., 2013a), Skip-Gram (Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and many others (Qiu et al., 2014; Niu et al., 2017; Peng and Zhou, 2020). BERT and its successor models can effectively generate contextual word embeddings with high-quality. However, the computational cost is very high. We reserve the study of contextual word embedding refinement as our future work. On the contrary, static word embedding models are generally simple and effective. These models assume that data distribution of words is in a linear structure. However, there still exists the situation that data distribution of words is in a strong non-linear structure (Chu et al., 2019), making the aforementioned models fail to estimate similarities between words. They may underestimate similarities between similar words and overestimate similarities between distant words, causing similarities obtained from word embedding models inconsistent with human judgment.

Some efforts have been made to address the inconsistency issue. For example, Locally Linear Embedding (LLE) method (Hasan and Curry, 2017) and Modified Locally Linear Embedding (MLLE) method (Chu et al., 2019) were proposed to refine pre-trained word vectors based on the weighted locally linear combination between words and their neighbors. The idea of these two similar methods is to utilize geometry information and keep the reconstruction weights between words and their local neighbors unchanged both in original and refined new embedding spaces. However, there are certain shortcomings in their methods. The reconstruction weights are constructed by the linear combination of words and their neighbors. These weights are easily influenced by different selections of neighboring words. Furthermore, the weighted linear combination used in their methods needs to perform twice and the whole process is time-consuming. The total operation needs to be performed separately in original and new embedding spaces.

In this paper, we propose two novel word representation refinement methods that overcome the shortcomings in previous methods. Our first Word Representation Refinement method utilizes Isometric Feature Mapping to refine word vectors based on the global geodesic distances between all words in the original embedding space (denoted as **WRR-IFM**). This method mainly focuses on global geometry information (geodesic distance) between all words. The geodesic distances between word points in the original embedding space are equal to those in a refined new embedding space through isometric feature mapping (Tenenbaum et al., 2000). The **WRR-IFM** method firstly computes the geodesic distances between all words by finding the shortest paths between them, then uses the isometric feature mapping method to re-embed word vectors from the original embedding space to a refined new embedding space. Meanwhile, we also introduce another novel Word Representation Refinement method by re-embedding word vectors based on Local Tangent Space (denoted as **WRR-LTS**). Our method considers a locally linear plane constructed by Principal Components Analysis (PCA) on word neighbors as an approximation of tangent space of each word. The tangent space of word points of manifold structure can represent the local geometry information (Zhang and Zha, 2002; Zhang and Zha, 2003). Then our **WRR-LTS** method re-embeds word vectors by aligning original and refined new embedding spaces based on the tangent space of each word. We conduct comprehensive experiments on seven different datasets with standard semantic relatedness and semantic similarity tasks to verify our proposed methods. The experimental results show that our **WRR-IFM** method can significantly refine the pre-trained word vectors and our **WRR-LTS** method achieves better performance than that of state-of-the-art baseline methods for word representation refinement. In summary, our contributions are presented as follows:

a) We introduce a word representation refinement method leveraging isometric feature mapping to correct word vectors based on the global geodesic distance between all words. This method mainly focuses on global geometry information (geodesic distance) between all words instead of local geometry information (the weighted locally linear combination between words) used previous studies.

b) We also introduce another word representation refinement method based on local tangent space. This method performs word representation refinement by aligning original and refined new embedding spaces based on different local geometry information, i.e. the local tangent space of words.

c) In this paper, we demonstrate that manifold-learning algorithms that preserve local geometry information are more beneficial to refine word representation in comparison with the manifold-learning algorithms that preserve global geometry information.

2 Related Work

Word representation is an essential component for semantic relatedness measurement in many IR tasks. In the past few years, different methods have been proposed to generate and refine word vectors.

Early idea about using vectors to represent words was derived from the vector space model (Salton et al., 1975), which utilized TF-IDF to construct a word-document co-occurrence matrix to represent words and documents as vectors. Subsequently, several methods were proposed to produce word embeddings by globally utilizing word-context co-occurrence counts based on word-context matrices in a corpus (Deerwester et al., 1990; Dhillon et al., 2011; Lebret and Collobert, 2013). These aforementioned methods all focus on word co-occurrence probability or word counts. These count-based methods do not consider the semantic relationships between words and their context words.

Apart from these count-based methods, there are prediction-based methods. These methods are derived from the distributed word representation hypothesis proposed by Hinton (1986). Distributed word representations represent words as dense and low-dimensional word vectors. There are many famous distributed word embedding models, such as C& W (Collobert et al., 2011), Continuous Bag-of-Words (CBOW) (Mikolov et al., 2013a), Skip-Gram (Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and many others (Qiu et al., 2014; Niu et al., 2017; Peng and Zhou, 2020). These methods leverage word context to generate word embeddings. Apart from the aforementioned static word embedding models, contextual embedding models become popular in recent days, such as BERT (Devlin et al., 2019), ELMO (Peters et al., 2018) and many others (Lan et al., 2020, Liu et al., 2020). These models demonstrate better performance on word embedding generation. In general, contextual word embedding models have a huge amount of parameters. Training such models are very time-consuming. On the contrary, static embedding models are far more simple and equally effective.

To improve the quality of word embeddings, many word representation refinement methods are proposed. Mu et al. (2018) post-processed pre-trained word vectors by removing the common mean vectors. Utsumi (2018) refined word vectors by using Layer-wise Relevance Propagation. Yu et al. (2017) utilized the ranking list of sentiment lexicon to guide word representation refinement. Methods utilizing manifold learning-based algorithms are particularly effective. Hasan and Curry (2017) proposed an method using Locally Linear Embedding (LLE) algorithm to re-embed pre-trained GloVe word vectors into a new embedding space. They used the weighted locally linear relationships between words and word neighbors in original space. Chu et al. (2019) used a Modified Locally Linear Embedding (MLLE) algorithm to refine pre-trained word vectors with the help of geometric information of words and neighboring words. Though they can achieve good performance, there still exist some limitations. The performance of above two manifold-learning based methods critically depends on local geometry information and the weighted locally linear combination between words and their (multiple) neighbors. The reconstruction weights are easily influenced by different selections of neighboring words. Also, the weighted locally linear combination needs to perform twice in both original and new embedding spaces. The whole process is quite time-consuming.

Because of these limitations, our **WRR-IFM** method tries to refine word representations by using global geometry information of all words instead of local geometry information. Our **WRR-LTS** method corrects word representations by using local geometry information (local tangent space) to align two embedding spaces rather than performing the weighted locally linear combination between words and neighboring words twice.

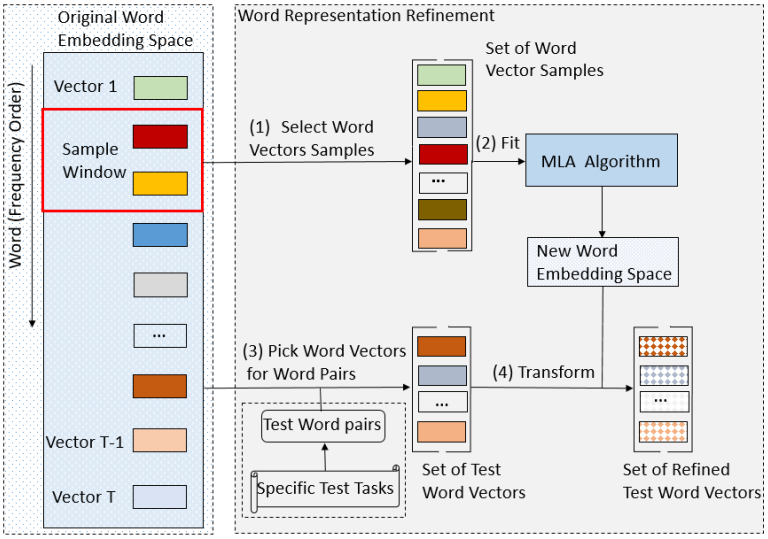

Figure 1. The framework of our proposed Word Representation Refinement Methods

3 Word Representation Refinement Methods

3.1 Overall Framework

Our proposed word representation refinement methods are based on a universal framework. The idea is to utilize manifold learning algorithms to re-embed word vectors from the original embedding space to a refined new embedding space. A sketch of the framework is shown in Figure 1. In the first step, we select a sample subset of word vectors from the original embedding space through a sample window. Word vectors are ordered by their corresponding word frequencies in a corpus. In this work, to demonstrate the effectiveness of our proposed methods, we test original word embeddings from GloVe[1], Word2Vec[2], and FastText[3]. Note that as in previous studies (Hasan and Curry, 2017; Chu et al., 2019), we use samples of word vectors rather than all vectors in the original embedding space to reduce high computational cost. In the second step, a fitted manifold learning algorithm will be used to transform word vectors from the original embedding space to a refined new embedding space with the dimension of word vectors retained. In the third step, we pick word vectors of word pairs in specific evaluation tasks from the original embedding space. Finally, we re-embed these word vectors to form new vectors in a new embedding space by the fitted manifold learning algorithm.

3.2 Word Representation Refinement based on Isometric Feature Mapping

LLE (Hasan and Curry, 2017) and MLLE methods (Chu et al., 2019) show promising results on word representation refinement. These two methods pay more attention to uncover the local geometry information of manifold structure. We make an attempt to exploit global geometry information instead of local geometry information of manifold structure. Hence, we propose a novel Word Representation Refinement method which utilizes Isometric Feature Mapping (**WRR-IFM**) to refine word vectors. The method is based on global geodesic distances between all words in the original embedding space. The basic assumption is that global geodesic distances between words are equal in original and refined new embedding spaces. In this method, we first compute global geodesic distances between all words by finding the shortest paths between them. Then we re-embed word vectors by applying the classical Multidimensional Scaling (MDS) technique to decompose the distance matrix constructed by geodesic distances.

1 https://nlp.stanford.edu/projects/glove/
2 https://code.google.com/archive/p/word2vec/
3 https://fasttext.cc/docs/en/english-vectors.html

Algorithm 1. Word Representation Refinement

Input: original word embedding space $\boldsymbol{S}$, test words $\{w_1, w_2, \cdots, w_m\}$
Output: refined vector set $\boldsymbol{Z}$ of test words

choose word vector samples $\boldsymbol{X} = [x_1, x_2, \cdots, x_N]$ from $\boldsymbol{S}$
For each $\boldsymbol{X} \in \boldsymbol{S}$ **do**
  If use **WRR-IFM**
    Fit $\boldsymbol{X}$ according to Eq. (2), (3) to obtain refined new word embeddings space $\boldsymbol{Y}$
  If use **WRR-LTS**
    Fit $\boldsymbol{X}$ according to Eq. (6), (10), (11) to obtain refined new word embeddings space $\boldsymbol{Y}$
end for
for all $w \in \{w_1, w_2, \cdots, w_m\}$ **do**
  obtain corresponding word vector of each w from $\boldsymbol{S}$
  re-embed word vector of w to obtain refined vector set $\boldsymbol{Z}$ of test words based on $\boldsymbol{Y}$
end for
return refined vector set $\boldsymbol{Z}$ of test words

We fit the Isometric Feature Mapping (IFM) algorithm on those selected samples. Firstly, we select word vector samples from the original embedding space $\boldsymbol{S}$ by using a sample window. The set of selected training samples is defined as a word vector set $\boldsymbol{X} = [x_1, x_2, \cdots, x_N]$, where N is the number of words. Note that $\boldsymbol{X} \in R^{d \times N}$, where d represents the dimension of word vectors. Then we fit the IFM algorithm based on $\boldsymbol{X}$. For each word vector point $x_i \in \boldsymbol{X}$, we find its k nearest neighbors (including x_i itself). Based on these neighbors, an undirected neighborhood graph G is constructed, where nodes represent word vectors (points) and edges represent links between two points. The edge weight between two neighboring points x_i and x_j in graph G is calculated by Euclidean distance $d_X(ij)$. If neighboring points x_i and x_j are linked, we initially set $d_G(ij) = d_X(ij)$, otherwise, $d_G(ij)$ is set to ∞. Graph G is updated by using the shortest path algorithm (Dijkstra algorithm). The shortest path from point x_i to x_j can be regarded as the geodesic distance between these two points:

$$d_G(ij) = \min\{d_G(ij), d_G(ik) + d_G(kj)\} \quad k = 1,2,\cdots N \tag{1}$$

The shortest path distances between all pairs of word vectors in graph G will form a matrix $\boldsymbol{D}_G$, where $\boldsymbol{D}_{ij} = d_G(ij)$. We use the classical MDS technique on $\boldsymbol{D}_G$ to re-embed word vectors into a new refined embedding space that can preserve the intrinsic geometry of the manifold structure. The re-embedded word vectors $y_i \in \boldsymbol{Y}$ for point x_i in refined space are chosen to minimize the cost function:

$$\arg min\, \boldsymbol{E} = ||\tau(\boldsymbol{D}_G) - \tau(\boldsymbol{D}_Y)||_{L^2} \tag{2}$$

Where $\boldsymbol{E}$ is the reconstruction error matrix, $\boldsymbol{D}_Y$ is the matrix of Euclidean distance $\{d_Y(ij) = ||y_i - y_j||\}$ in new refined space and $||\boldsymbol{A}||_{L^2}$ is the L^2 matrix norm $\sqrt{\sum_{i,j} A_{i,j}^2}$ ($\boldsymbol{A}$= $\tau(\boldsymbol{D}_G) - \tau(\boldsymbol{D}_Y)$). The τ operator converts distance to inner products, which uniquely characterize the geometry of data in a form that supports efficient optimization. The global minimum of Eq. (2) can be achieved by setting the vector y_i in a refined word embedding set $\boldsymbol{Y}$ (which is also regarded as refined new embedding space) to the top t eigenvectors of the matrix $\tau(D_G)$, where t is equal to d, as the embedding dimension is identical in original and new embedding space. To obtain these eigenvectors, we compute $(\boldsymbol{D}_G) = -\frac{1}{2}\boldsymbol{G}\boldsymbol{D}_G\boldsymbol{G}^T$, where $\boldsymbol{G}$ is a Householder centering matrix. Then we compute eigenvalue decomposition $\tau(\boldsymbol{D}_G) = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ with $\lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, were $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Finally, we choose top t nonzero eigenvalues and corresponding eigenvectors as refined word embedding coordinates. The refined word embedding set $\boldsymbol{Y}$ can be obtained by Eq. (3), which is computed by Eq. (4) and Eq. (5) below:

$$\boldsymbol{Y} = \boldsymbol{U}_t \Lambda_t^{\frac{1}{2}} \tag{3}$$

$$\boldsymbol{U}_t = [u_1, \cdots, u_t], u_i \in R^n \tag{4}$$

$$\Lambda_t = diag(\lambda_1, \ldots, \lambda_t), \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0 \tag{5}$$

According to Eq. (2) and Eq. (3), we train the IFM algorithm on selected word vector training samples to obtain a refined new embedding space $\boldsymbol{Y}$. Then, we pick the test word vectors from the original embedding space and re-embed them to obtain refined vector set of test words by leveraging the new embedding space $\boldsymbol{Y}$. The overall procedure of our proposed Word Representation Refinement methods for both **WRR-IFM** (as well as **WRR-LTS**) is described in Algorithm 1.

3.3 Word Representation Refinement based on Local Tangent Space

Similar to LLE (Hasan and Curry, 2017) and MLLE methods (Chu et al., 2019), our **WRR-LTS** method also considers preserving local geometry information of words and their neighbors for refining word vectors. However, local geometry information used in our method is different from those of LLE and MLLE methods. Furthermore, to overcome the limitations of these two methods, our method utilizes local tangent space of word points instead of performing weighted locally linear combination twice for word representation refinement. In this proposed method, we firstly construct a locally linear plane by utilizing PCA on words and their neighbors. This plane is regarded as an approximation of the tangent space at each word. Due to the existence of a linear mapping of each word from both original and new embedding spaces to its local tangent space, our method re-embeds word representations by aligning these linear mappings based on this local tangent space.

The procedure of the proposed method described in this section is similar to that of the **WRR-IFM** method. We firstly select word vector samples from the original embedding space $\boldsymbol{S}$ via a sample window and this selected word vector sample set is defined as $\boldsymbol{X} = [x_1, x_2, \cdots, x_N]$. Note that $\boldsymbol{X} \in R^{d \times N}$, where d and N represent the dimension of word vector samples and the number of word vectors. Then we train a Local Tangent Space (LTS) algorithm on them. To be specific, for each word vector $x_i \in \boldsymbol{X}$, we need to find its k nearest neighborhoods (including x_i itself) and the adjacent neighborhood set is denoted as $\boldsymbol{X}_i = [x_{i1}, x_{i2}, \cdots, x_{ik}]$. To preserve the local structure of the neighborhood set $\boldsymbol{X}_i$ of each word vector x_i, we apply PCA to $\boldsymbol{X}_i$ to approximate the local tangent space of the word corresponding to a word vector x_i. The objective function is

$$arg\min_{\boldsymbol{Q}_i, \theta_i} \sum_{j=1}^{k} ||(x_{ij} - x) - \boldsymbol{Q}\theta_{ij}||^2 = arg\min_{\boldsymbol{Q}_i, \boldsymbol{\Omega}_i} ||\boldsymbol{X}_i(\boldsymbol{I} - \frac{ee^T}{k}) - \boldsymbol{Q}\boldsymbol{\Omega}_i||^2 \tag{6}$$

where $\boldsymbol{I}$ is an identity matrix, e represents the vector of all 1's, $\boldsymbol{Q}$ is an orthonormal basis matrix of the tangent space, $\boldsymbol{\Omega}_i = [\theta_{i1}, \theta_{i2}, \cdots, \theta_{ik}]$ represents the local linear approximation of $\boldsymbol{X}_i$. The optimal x in the above formula is given by neighborhood set $\boldsymbol{X}_i$, because it is the mean value of all word vectors x_{ij}, $(j = 1,2, \cdots k)$ in $\boldsymbol{X}_i$. The optimal $\boldsymbol{Q}$ is given by the orthogonal basis $\boldsymbol{Q}_i$ and is composed of t left singular vectors of $\boldsymbol{X}_i(\boldsymbol{I} - \frac{ee^T}{k})$ corresponding to its t largest singular values ($t = d$ and the reason is mentioned in Section 3.2). The tangent coordinates $\boldsymbol{\Omega}_i$ can be defined as

$$\boldsymbol{\Omega}_i = \boldsymbol{Q}_i{}^T \boldsymbol{X}_i(\boldsymbol{I} - \frac{ee^T}{k}) \tag{7}$$

After we extract local tangent coordinates $\boldsymbol{\Omega}_i$ by an optimal linear fitting to neighboring samples, we need to obtain the global coordinates in new embedding space. The purpose of global coordinate construction is to find a group of global coordinates in a new embedding space. We assume that there is an alignment matrix, which re-embeds tangent coordinates $\boldsymbol{\Omega}_i$ to new space coordinates $\boldsymbol{Y}_i = \{y_{i1}, y_{i2}, \cdots, y_{iN}\}$ in new embedding space, then we have

$$\boldsymbol{Y}_i(\boldsymbol{I} - \frac{ee^T}{k}) = \boldsymbol{L}_i\boldsymbol{\Omega}_i + \boldsymbol{E}_i \tag{8}$$

where $\boldsymbol{L}_i$ is the alignment matrix which maps $\boldsymbol{\Omega}_i$ to $\boldsymbol{Y}_i$ and $\boldsymbol{E}_i$ is the local reconstruction error matrix. To preserve as much of local geometry information in the new embedding space as possible, we seek to find $\boldsymbol{Y}_i$ and $\boldsymbol{L}_i$ to minimize the reconstruction error $\boldsymbol{E}_i$

$$\arg\min_{\boldsymbol{Y}} \sum_{i=1}^{N} ||\boldsymbol{E}_i||^2 = \arg\min_{\boldsymbol{Y}} \sum_{i=1}^{N} ||\boldsymbol{Y}_i(\boldsymbol{I} - \frac{ee^T}{k}) - \boldsymbol{L}_i\boldsymbol{\Omega}_i||^2 \tag{9}$$

Obviously, the optimal alignment matrix $\boldsymbol{L}_i$ has the form $\boldsymbol{L}_i = \boldsymbol{Y}_i(\boldsymbol{I} - \frac{ee^T}{k})\boldsymbol{\Omega}_i^+$, and the local reconstruction error $\boldsymbol{E}_i = \boldsymbol{Y}_i(\boldsymbol{I} - \frac{ee^T}{k})(\boldsymbol{I} - \boldsymbol{\Omega}_i^+\boldsymbol{\Omega}_i)$ is minimal, where $\boldsymbol{\Omega}_i^+$ is Moore-Penrose generalized inverse of $\boldsymbol{\Omega}_i$. Let refined word vector set $\boldsymbol{Y} = [y_1, y_2, \cdots, y_N]$ in new embedding space ($\boldsymbol{Y}$ is also called as refined new embedding space) and $\boldsymbol{\psi}_i$ be the 0-1 selection matrix such that $\boldsymbol{Y}\boldsymbol{\psi}_i = \boldsymbol{Y}_i$. We find the optimal $\boldsymbol{Y}$ by minimizing the overall reconstruction error and the objective function in Formula (9) can be rewritten as:

$$arg\min_{\boldsymbol{Y}} \sum_{i} ||\boldsymbol{E}_i||_F^2 = arg\min_{\boldsymbol{Y}} ||\boldsymbol{Y}\boldsymbol{\psi}\boldsymbol{W}||_F^2 = \min trace\,(\boldsymbol{Y}\boldsymbol{B}\boldsymbol{Y}^{\boldsymbol{T}}) \tag{10}$$

where $\boldsymbol{\psi} = [\psi_1, \psi_2, \cdots, \psi_N]$, $\boldsymbol{W} = diag\,(W_1, W_2, \cdots, W_N)$ with $\boldsymbol{W}_i = (\boldsymbol{I} - \frac{ee^T}{k})(\boldsymbol{I} - \boldsymbol{\Omega}_i^+\boldsymbol{\Omega}_i)$ and $\boldsymbol{B} = \boldsymbol{\psi}\boldsymbol{W}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{\psi}^{\boldsymbol{T}}$. To uniquely obtain $\boldsymbol{Y}$, the constraint $\boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}} = \boldsymbol{I}$ is imposed. The vector e of all ones is an eigenvector of $\boldsymbol{B}$ corresponding to a zero eigenvalue. Then the refined word embedding set $\boldsymbol{Y}$ is given by the t eigenvectors of the matrix $\boldsymbol{B}$, corresponding to the 2nd to $(t+1)$th smallest eigenvalues of $\boldsymbol{B}$, and the eigenvector matrix picked from $\boldsymbol{B}$ is $[u_2, \cdots, u_{t+1}]$, where u_i is an eigenvector of $\boldsymbol{B}$. Then d dimensional refined new embedding set $\boldsymbol{Y}$ should be:

$$\boldsymbol{Y} = [u_2, \cdots, u_{t+1}] \tag{11}$$

We use word vector samples from the original embedding space to train the LTS algorithm by Eq. (6), Eq. (10) and Eq. (11) to obtain a refined new embedding space $\boldsymbol{Y}$. Then we obtain word vectors of test words from the original embedding space and obtain refined vector set of these words based on the new embedding space $\boldsymbol{Y}$.

4 Experiments and Results

4.1 Datasets

We utilize two semantic relatedness and similarity tasks to validate the performance of our proposed word representation refinement methods. The semantic relatedness task contains two datasets, including WordRel (WordRel) dataset (252 noun pairs) (Agirre et al., 2009), and MTurk (MTurk) dataset (287 word pairs) (Kira et al., 2011). The semantic similarity task includes five datasets, which are RG65 (RG) dataset (65 noun pairs) (Rubenstein et al., 2000), WordSim-353 (WS353) dataset (353 noun pairs) (Finkelstein et al., 2001), SimLex-999 (SimLex) dataset (999 word pairs) (Hill and Korhonen, 2015), SimVerb-3500 (SimVerb) dataset (3500 verb pairs) (Gerz et al., 2016) and WordSim-203 (WS203) dataset (203 noun pairs) (Gerz et al., 2016) respectively.

We use three types of pre-trained word vectors in our word representation refinement experiments, which are GloVe (Pennington et al., 2014), FastText (Mikolov et al., 2018) and Word2Vec (Mikolov et al., 2013a). GloVe word vectors are learned from different sources. 400,000 GloVe vectors are trained on Wikipedia 2014+Gigaword 5 corpora (consists of 6 Billion tokens, 400,000 vocabularies, word vectors with 50, 100, 200, and 300 dimensions). Another 1.9 Million GloVe vectors are trained on Common Crawl corpus (consists of 42 Billion tokens, 1.9 Million vocabularies, word vectors with 300 dimensions). 1 million FastText vectors are trained on Wikipedia 2017 corpus (consists of 16 Billion tokens, word vectors with 300 dimensions, 1 million words). Word2vec vectors with 300 dimensions are trained on part of Google News dataset, consisting of 3 million words and phrases.

4.2 Baselines and Evaluation Metrics

We report our experimental results in comparison with other state-of-the-art word representation refinement methods. The description of baseline methods is presented as follows.

GloVe. GloVe (Pennington et al., 2014) vectors are trained on global word co-occurrence statistics and it considers both global and local features of words.

Word2Vec. Word vectors trained by Word2Vec model (Mikolov et al., 2013a) only focus on local features of words. This method utilizes the sliding context windows to select neighboring words to predict the target word, or the current word to predict its neighbors.

FastText. This method (Mikolov et al., 2018) considers subwords and uses them to deal with out-of-vocabularies when producing word vectors.

LLE. This method is proposed in Hasan and Curry (2017)'s work to preserve local linear features between words and their neighbors by using the LLE manifold learning algorithm.

MLLE. Similar to LLE descried above, Chu et al. (2019) used the MLLE manifold learning algorithm to refine pre-trained word vectors.

WRR-ISM. The first method we proposed in this paper. We use an Isometric Feature Mapping algorithm that focuses on preserving geodesic distance between words to re-embed word vectors from the original embedding space to a refined new embedding space.

WRR-LTS. The second method we proposed in this paper. It uses the Local Tangent Space algorithm to re-embed word vectors by aligning original and refined new space based on the tangent space of each word.

4.3 Results and Discussion

4.3.1 Performance on Two Evaluation Tasks

We conduct experiments on seven datasets of semantic similarity and semantic relatedness tasks to verify the performance of our proposed methods. Table 1 shows the comparison results of our proposed methods (**WRR-IFM** and **WRR-LTS**) and all baseline methods (LLE and MLLE) on three different sets of pre-trained word vectors.

The results show that, when all methods are trained on GloVe vectors, the **WRR-LTS** method achieves the best scores on five out of seven datasets. When they are trained on Word2Vec vectors, the **WRR-LTS** method achieves the best scores on five out of seven datasets. When they are trained on the FastText vectors, the **WRR-LTS** method achieves the best scores on six out of seven datasets. The clear advantage of the **WRR-LTS** method demonstrates that our local tangent space-based method captures more accurate local geometry information than those of baseline LLE and MLLE methods. In other words, local tangent space in our proposed method is more beneficial to represent the local geometry information in comparison with weighted locally linear combination between words and their neighbors in LLE and MLLE methods.

However, our **WRR-IFM** method works less well. In most of runs, it can only bring improvements over the original word embeddings (i.e. GloVe, word2ve and FastText) and fail to outperform various baseline methods. This shows that local geometry information may be more important than global geometry information in refining word representations. Global geometry information may introduce some noises in the whole refining process.

We now examine the differences between two evaluation tasks. We can see that all manifold learning-based methods including our proposed methods demonstrate similar performance. These results are inline with previous findings, so that the two tasks are quite suitable to evaluate the word representation refinement.

4.3.2 Comparison of Refining Different Word Vectors

In Hasan and Curry (2017) and Chu et al., (2019)'s work, they only compared manifold learning-based methods with the GloVe vectors. In this paper, we try to compare their performance on three representative and popular pre-trained word vectors. From Table 1, we can see that our method (**WRR-LTS)** outperforms GloVe with improvements from 1.77% to 10.22%, outperforms FastText with

	GloVe pre-trained word vectors						
	Semantic Similarity					Semantic Relatedness	
	RG	WS353	SimLex	SimVerb	WS203	MTurk	WordRel
Glove	76.90	71.25	40.83	28.33	80.15	69.29	64.43
LLE	74.71	77.14	48.14	36.55	81.40	71.92	72.90
MLLE	77.19	78.40	49.40	37.32	82.32	72.78	73.69
WRR-IFM	**83.38†**	67.23	**41.08†**	27.28	71.23	**71.27†**	**64.45†**
WRR-LTS	**86.48***	**78.78***	**50.46***	**35.74†**	**81.92†**	**73.15***	**74.65***
	Word2Vec pre-trained word vectors						
	Semantic Similarity					Semantic Relatedness	
	RG	WS353	SimLex	SimVerb	WS203	MTurk	WordRel
Word2Vec	77.23	65.25	45.39	37.54	76.67	71.71	59.20
LLE	81.65	69.78	46.55	38.80	77.01	70.61	62.94
MLLE	82.11	69.67	47.03	38.84	78.36	70.97	66.23
WRR-IFM	**80.49†**	**67.29†**	44.50	34.11	**78.82†**	**71.95***	**65.21†**
WRR-LTS	**83.46***	**70.09***	**47.05***	**38.86***	**79.77***	71.01	**66.14†**
	FastText pre-trained word vectors						
	Semantic Similarity					Semantic Relatedness	
	RG	WS353	SimLex	SimVerb	WS203	MTurk	WordRel
FastText	82.63	70.29	46.97	37.28	80.09	72.09	67.00
LLE	86.16	74.11	47.68	40.80	81.47	73.41	69.97
MLLE	87.59	76.95	49.40	41.02	83.04	73.54	73.88
WRR-IFM	**87.09†**	**71.86†**	**47.44†**	**40.43†**	**80.50†**	**74.20***	**67.13†**
WRR-LTS	**90.15***	**77.64***	**49.53***	**41.08***	**84.43***	**74.00†**	**74.29***

Table 1: Spearman correlations between scores predicted by our model and scores obtained from human judgment on seven specific datasets. Bold values with * represent our proposed approach achieve the best performance among all baseline methods. Bold values with † represent our proposed method achieves better results than original pre-trained models. Note that all baseline results of GloVe pre-trained word vectors are taken from the study (Chu et al., 2019)

improvements from 1.45% to 6.88%, outperforms Word2Vec with improvements from 1.32% to 6.94%. The larger improvement on count-based model (GloVe) than prediction-based model needs further investigation, we leave it as our future work.

4.3.3 Performance on Refining GloVe Word Vectors

To compare the performance of different embedding dimensions of GloVe word vectors of our proposed methods and all baseline methods, we randomly choose two datasets, WS353 and RG to report results. The results are shown in Table 2. Compared with baseline methods (including LLE and MLLE), our **WRR-LTS** method achieves the best performance in 5 out of 10 experimental runs and our **WRR-IFM** method obtains the highest scores in 4 out of 10 experimental runs. Our methods also show significant improvement in most of runs in comparison with original GloVe word vectors. When the dimension and training size increase, the performance is better. So that in section 4.3.1, all GloVe vectors are trained with the most data that we can obtain.

4.3.4 Impact of Parameters

Finally, we describe the impact of all parameters. We set the number of eigenvectors to be equal to the dimension of pre-trained word vectors. The size of training sample window is in the range [300, 1000]. The value range of number of neighbors is chosen from [300, 1500]. Generally, the lower number of local neighbors is, the faster the fitted manifold learning algorithm runs.

Space	Task	GloVe	LLE	MLLE	**WRR-IFM**	**WRR-LTS**
6B50d	WS353	61.2	56.6	63.2	58.7	61.2
6B50d	RG	60.2	53.0	64.4	**67.6***	**62.6†**
6B100d	WS353	64.5	64.3	64.6	64.1	**66.4***
6B100d	RG	65.3	67.3	68.8	**72.5***	**73.3***
6B200d	WS353	68.5	69.7	67.0	65.4	68.2
6B200d	RG	75.5	76.0	79.4	**77.1†**	**81.5***
6B300d	WS353	65.8	70.3	67.9	**67.2†**	**69.3†**
6B300d	RG	75.5	80.5	81.1	**83.4***	**83.1†**
42B300d	WS353	75.2	78.4	78.6	72.1	**78.8***
42B300d	RG	80.0	83.4	83.5	**84.5***	**86.5***

Table 2: Spearman correlations between scores predicted by our model and scores obtained from human judgment on two evaluation datasets. Bold values with * represent our proposed approach achieve the best performance among all baseline methods. Bold values with † represent our proposed method achieves better results than the original GloVe pre-trained model. Note that baseline results are taken from the study (Chu et al., 2019)

5 Conclusion and Future Work

In this paper, we study word representation refinement problem by utilizing manifold learning algorithms. We propose two novel methods (**WRR-IFM** and **WRR-LTS**) for this purpose. Our **WRR-IFM** method utilizes isometric feature mapping to refine word vectors based on the global geodesic distance between all words in the original embedding space. Our **WRR-LTS** method corrects word representations by aligning original and refined new embedding space based on the tangent space of words. The **WRR-IFM** method focuses on preserving global geometry information (global geodesic distances) between all words, while our **WRR-LTS** method considers local geometry information (local tangent space of words) between words and their neighbors. We conduct several experiments on semantic relatedness and semantic similarity tasks. The results obtained in these two evaluation tasks suggest that our proposed methods consistently perform well for refining word representations. In the future, we intend to extend our experiments to refine aligned bilingualism and multilingual word vectors. We also intend to investigate whether our proposed methods have a significant impact on refining contextual word embeddings.

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.

Reference

Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, Aitor Soroa. 2009. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 19-27, Colorado, USA.

Ebrahim Bagheri, Faezeh Ensan, and Feras Al-Obeidat. 2018. Neural word and entity embeddings for ad hoc retrieval. *Information Processing & Management*, 54(4): 657-673.

Yonghe Chu, Hongfei Lin, Liang Yang, Yufeng Diao, Shaowu Zhang, Xiaochao Fan. 2019. Refining word representations by manifold learning. *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages: 5394-5400, Macao, China.

Ronan Collobert, Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning (ICML)*, pages: 160-167, Helsinki, Finland.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, & Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12(1):2493-2537.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), pages: 391-407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages: 4171-4186, Minneapolis, MN, USA.

Srikanth Mujjiga, Vamsi Krishna, Kalyan Chakravarthi, and J. Vijayananda. 2019. Identifying semantics in clinical reports using neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, pages: 9552-9557, Hawaii, USA.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view Learning of Word Embeddings via CCA. *Advances in neural information processing systems (NIPS)*, pages: 199-207.

Massimo Esposito, Emanuele Damiano, Aniello Minutolo, Giuseppe De Pietro, and Hamido Fujita. 2020. Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, 514: 88-105.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias. 2001. Placing search in context: The concept revisited. *Proceedings of the 10th international conference on World Wide Web (WWW)*, pages 406-414, Hong Kong, China.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages:2173-2182, Texas, USA.

Souleiman Hasan, Edward Curry. 2017. Word Re-Embedding via Manifold Dimensionality Retention. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages: *321-326*.

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

Geoffrey E. Hinton. 1986. Learning distributed representations of concepts. *Proceedings of the eighth annual conference of the cognitive science society*, vol 1, pages: 1-12.

Radinsky Kira, Eugene Agichtein, Evgeniy Gabrilovich. 2011. A word at a time: computing word relatedness using temporal semantic analysis. *Proceedings of the 20th international conference on World Wide Web (WWW)*, pages 337–346, Hyderabad, India.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, pages: 1-17, Addis Ababa, Ethiopia.

Rémi Lebret and Ronan Collobert. 2013. Word emdeddings through hellinger PCA. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Idiap, pages: 482-490, Gothenburg, Sweden.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, Qi Ju. 2020. FastBERT: a Self-distilling BERT with Adaptive Inference Time. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages: 6035-6044, Online.

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, pages: 3111-3119, Arizona, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems (NIPS)*, pages: 3111-3119.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pages: 52-55, Miyazaki, Japan.

Jiaqi Mu, Suma Bhat, Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. *Proceedings of Poster at 6th International Conference on Learning Representations (ICLR)*, pages: 1-25, Vancouver, Canada.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, Maosong Sun. 2017. Improved Word Representation Learning with Sememes. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, (1): 2049-2058, Vancouver, Canada.

Dwaipayan Roy, Debasis Ganguly, Sumit Bhatia, Srikanta Bedathur, Mandar Mitra. 2018. Using word embeddings for information retrieval: How collection and term normalization choices affect performance. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pages: 1835-1838, Torino, Italy.

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

Gerard Salton, Anita Wong, Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11): 613-620.

Roger A. Stein, Patricia A. Jaques, and Joao Francisco Valiati. 2019. An analysis of hierarchical text classification using word embeddings. *Information Sciences*, 471: 216-232.

Joshua B. Tenenbaum, Vin De Silva, John C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319-2323.

Joseph Turian, Ratinov Lev, Bengio Yoshua. 2010. Word representations: a simple and general method for semi-supervised learning. *Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics (ACL)*, pages: 384-394, Uppsala, Sweden.

Ann A. O'Connell. 1999. Modern multidimensional scaling: theory and applications. *Journal of the American Statistical Association*, 94(445): 338-340.

Xiaoya Peng, Dong Zhou. 2020. A Framework for Learning Cross-Lingual Word Embedding with Topics. *Proceedings of Web and Big Data - 4th International Joint Conference (APWeb-WAIM)*, pages: 285-293, Tianjin, China.

Jeffrey Pennington, Richard Socher, Christopher D. Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages: 1532-1543, Doha, Qatar.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages: 2227–2237, Louisiana, USA.

Akira Utsumi. 2018. Refining Pretrained Word Embeddings Using Layer-wise Relevance Propagation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages: 4840-4846, Brussels, Belgium.

Lin Qiu, Yong Cao, Zaiqing Nie, Yong Yu, Yong Rui. 2014. Learning word representation considering proximity and ambiguity. *Proceedings of Twenty-eighth AAAI conference on artificial intelligence (AAAI)*, pages: 1572-1578, Québec, Canada.

Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. *Proceedings of the 2017 conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages: 534-539, Copenhagen, Denmark.

Zhenyue Zhang, Hongyuan Zha. 2002. Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. *Journal of Shanghai University*, 8(4), pages: 406-424.

Zhenyue Zhang, Hongyuan Zha. 2003. Nonlinear Dimension Reduction via Local Tangent Space Alignment. *Proceedings of International Conference on Intelligent Data Engineering and Automated Learnin*g (IDEAL), pages: 477-481, Hong Kong, China.