Tian F, Gao B, Chen EH *et al.* Learning better word embedding by asymmetric low-rank projection of knowledge graph. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(3): 624–634 May 2016. DOI 10.1007/s11390-016-1651-5

Learning Better Word Embedding by Asymmetric Low-Rank Projection of Knowledge Graph

Fei Tian[1], Bin Gao[2], *Member, ACM, IEEE*, En-Hong Chen[1], *Senior Member, IEEE*, and Tie-Yan Liu[2], *Senior Member, CCF, ACM, IEEE*

[1]*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China*
[2]*Microsoft Research Asia, Beijing 100080, China*

E-mail: tianfei@mail.ustc.edu.cn; bingao@microsoft.com; cheneh@ustc.edu.cn; tyliu@microsoft.com

Received November 9, 2015; revised February 25, 2016.

Abstract Word embedding, which refers to low-dimensional dense vector representations of natural words, has demonstrated its power in many natural language processing tasks. However, it may suffer from the inaccurate and incomplete information contained in the free text corpus as training data. To tackle this challenge, there have been quite a few studies that leverage knowledge graphs as an additional information source to improve the quality of word embedding. Although these studies have achieved certain success, they have neglected some important facts about knowledge graphs: 1) many relationships in knowledge graphs are many-to-one, one-to-many or even many-to-many, rather than simply one-to-one; 2) most head entities and tail entities in knowledge graphs come from very different semantic spaces. To address these issues, in this paper, we propose a new algorithm named ProjectNet. ProjectNet models the relationships between head and tail entities after transforming them with different low-rank projection matrices. The low-rank projection can allow non one-to-one relationships between entities, while different projection matrices for head and tail entities allow them to originate in different semantic spaces. The experimental results demonstrate that ProjectNet yields more accurate word embedding than previous studies, and thus leads to clear improvements in various natural language processing tasks.

Keywords natural language processing, word embedding, neural network, knowledge graph

1 Introduction

In recent years, the research on word embedding (or distributed word representations) has made promising progress in many natural language processing tasks[1-6]. Different from traditional one-hot discrete representations of words, word embedding vectors are dense, continuous, and low-dimensional. They are usually trained with neural networks on a large-scale free text corpus, such as Wikipedia, news articles, and web pages, in an unsupervised manner.

While word embedding has demonstrated its power in many circumstances, it is gradually recognized that conventional word embedding techniques may suffer from the incomplete and inaccurate information contained in the free text data. On one hand, due to the restrictive topics and coverage of a text corpus, some words might not have sufficient contexts and therefore might not have reliable word embeddings. On the other hand, even if a word has sufficient contextual data, the free texts might be inaccurate and thus might not provide a semantically precise view of the word. As a result, the learned word embedding might be unable to carry on the desirable semantic information. To tackle this problem, recently some researchers have proposed to leverage knowledge graphs, such as WordNet[7] and Freebase[8], as additional data sources to improve word embedding[9-11].

In summary, these studies believe that knowledge graphs are helpful in generating better word embedding vectors, given that knowledge graphs are usually built and validated by human experts' efforts and thus have much better quality in reflecting the inherent relationships between words than free text corpus. In this way, the noise and bias in free text corpus can be alleviated by leveraging knowledge graphs. In addition, cur-

Regular Paper

rent knowledge graphs have very good coverage of daily words/entities[12], leading to their significant overlaps with large text corpus and furthermore a substantial improvement over free text corpus alone.

A knowledge graph contains a set of nodes representing entities and a set of edges corresponding to the relationships between entities. In other words, a knowledge graph can be regarded as a set of triples (h, r, t), where head entity h and tail entity t share relationship r. In [10-11], in addition to the original likelihood loss on the free texts, an extra loss function is imposed to capture the relationships in the knowledge graph. Specifically, the additional loss takes the form $L_{\mathcal{K}} = \sum_{(h,r,t)} ||\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}||_2^2$, where $\boldsymbol{h}, \boldsymbol{t}$ are the embedding vectors of the words (entities) h and t respectively, and $\boldsymbol{r}$ is the embedding vector of the relationship r. Then the embeddings are learned by minimizing the overall loss on both free text and knowledge graph.

While the above approaches have shown certain success, we would like to point out their limitations.

First, the loss function $L_{\mathcal{K}}$ in these studies cannot capture complex relationships between entities. In particular, it will encounter problems when the relationships are one-to-many, many-to-one, or many-to-many. For example, r = "cause of death" is a many-to-one relationship, since many different head entities h_i (e.g., h_1 = "Abraham Lincoln" and h_2 = "John F Kennedy") correspond to the same tail entity (e.g., t = "assassination by firearm"). In the case, the minimization of $L_{\mathcal{K}}$ will enforce the embedding vectors of all head entities (e.g., $\boldsymbol{h_1}$, $\boldsymbol{h_2}$) to approach each other, which is clearly unreasonable.

Actually such kind of complex relationships are very common in knowledge graphs. Take a widely used benchmark dataset $FB13$ [13], which is a subset of Freebase, as an instance. For every relationship in $FB13$, we calculate the average number of head entities corresponding to one tail entity and the average number of tail entities corresponding to one head entity. Then we obtain the means and standard deviations of such values under different relationships. The overall statistical information is listed in Table 1, from which we can see that relationships in $FB13$ are highly non one-to-one, especially for the mapping from tail entity to head entity, as shown by the large mean value of *#Head per Tail* (the number of head entities per tail entity). In addition, the standard deviation for *#Head per Tail* is fairly large, indicating that the degrees of non one-to-one mappings from tail entity to head entity vary drastically across different relationships. This clearly shows that the issue is very serious and we should tackle it in order to learn a reasonable word embedding.

Table 1. Number of Head Entities per Tail Entity and Number of Tail Entities per Head Entity in $FB13$

	Mean	Std. Deviation
#Head per Tail	2 614.17	9 229.75
#Tail per Head	1.26	0.23

Second, the loss function $L_{\mathcal{K}}$ adopts simple arithmetic operations on the embedding vectors of the head and the tail entities, implying that both entities are located in the same space. However, the fact is that head entities are usually more concrete and tail entities are more abstract, making it unreasonable to simply regard them as in a homogeneous space. Still using the above example, for relationship r = "cause of death", all the head entities are real human names whereas all the tail entities are abstract reasons of death. What is more, according to Table 1, head and tail entities are not symmetric from the statistics perspective: the number of tail entities per head entity is much smaller than that of head entities per tail entity, further indicating the heterogeneity nature of head and tail entities and suggesting that we should treat them separately in the mathematical modeling.

In the literature, there are some research studies that try to resolve one of the aforementioned issues. However, as far as we know, none of the studies successfully addressed both issues. For example, in [14], it is proposed to project the embedding vectors of both entities onto a relation-dependent hyperplane before computing the loss function $L_{\mathcal{K}}$. However, the heterogeneity between head and tail entities is not considered. Furthermore, the projection matrix used in [14] has a fixed rank for all types of relationships, which could not express various degrees of non one-to-one mappings. In [15], different transformations are adopted to head and tail entities respectively. However, no consideration is taken to address the issue of non one-to-one mappings.

To address the limitations of existing studies, in this paper, we propose a new algorithm called ProjectNet, which adopts different and carefully designed projections to the head and the tail entities respectively when defining the loss function $L_{\mathcal{K}}$. First, we show that the necessary condition to resolve the issue of non one-to-one mapping is to ensure the projection matrix to be low-rank. In such a way, we can guarantee the translation distance between the entities to be small after projection without forcing their embedding vectors to be the same. Actually, it can be proven that the TransH model[14] is our special case in the sense that it also adopts a projection matrix of low (and fixed) rank. Our model is more general since we can explicitly control the rank of the projection matrix to adapt to knowledge graphs with different degrees of non one-to-one mappings. Second, by using different projection matrices for head and tail entities respectively, we can avoid the homogeneity assumption on the semantic space and therefore build a more flexible and accurate model. For example, for the knowledge graph $FB13$, we should adopt a low-rank projection matrix for head entities since the number of head entities is very large for each tail entity; however, it is safe to use a relatively full rank projection matrix for tail entities since the number of tail entities is rather small for each head entity.

We have tested the performance of our proposed algorithm on several benchmark datasets, and the experimental results show that our proposal can significantly outperform the baseline methods. This indicates the benefit of carefully modeling entities and relationships when incorporating knowledge graphs into the learning process of word embedding.

The rest of the paper is organized as following. In Section 2, we summarize related work in leveraging knowledge graph to help word embedding. Then in Section 3, the detailed model is introduced and its difference with related methods is illustrated. After that, the experimental settings and results are reported in Section 4. The paper is finally concluded in Section 5.

2 Related Work

Word embeddings (a.k.a. distributed word representations), are usually trained with neural networks by maximizing the likelihood of a text corpus. Based on several pioneering efforts[1-2,6], the research studies in this field have grown rapidly in recent years[3-6,17-18]. Among them, *word2vec*[4-5] draws quite a lot of attention from the community due to its simplicity and effectiveness. An interesting result given by *word2vec* is that the word embedding vectors it produces can reflect human knowledge via some simple arithmetic operations, e.g., $v(Japan) - v(Tokyo) \approx v(France) - v(Pairs)$.

However, as aforementioned, word embedding models like *word2vec* usually suffer from the incompleteness and inaccuracy of the free-text training corpus. To address this challenge, there are some attempts that leverage additional structured or unstructured human knowledge to enhance word embeddings. Here are some examples. In [19-20], the authors adopted morphological knowledge to aid the learning of rare words and new words. In [21], the authors used semantic relational knowledge between words as a constraint in learning word embedding vectors. In [11], the authors leveraged knowledge graphs, the most widely used structured knowledge, to help improve word representations. In particular, the authors[11] did not only minimize the loss on the text corpus, but also minimize the loss on the knowledge graph by sharing embedding vectors between words and entities. In [10], the authors proposed a very similar method to [11], but with a different objective of improving knowledge graph understanding with the help of text corpus. Actually, both the models in [11] and [10] are inspired by the TransE model[22], which is the state-of-the-art work in the literature of computing distributed representations for knowledge graphs[13,15,23]. In TransE, the relational operation between entities h, t with relationship r is assumed to be a simple linear translation, i.e., $\min ||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_2^2$. However, as pointed out in the introduction, such a simple formulation cannot handle the non one-to-one mappings between entities. To tackle the problem, in [14], the authors proposed a simple projection method named TransH. We will review the detailed mathematical forms of these models in Subsection 3.3 and discuss their relationship with our proposal.

3 ProjectNet Algorithm

In this section, we introduce our proposed ProjectNet model in details. In general, following [10-11], given a training text corpus $\mathcal{D}$ and a set $\mathcal{K}$ of triples in the form (*head entity, relation, tail entity*) extracted from a knowledge graph, our model jointly minimizes a linear combination of the loss items on both text and knowledge:

$$L = (1 - \alpha)L_{\mathcal{D}} + \alpha L_{\mathcal{K}}, \tag{1}$$

where $\alpha \in [0, 1]$ is used to trade off the two loss terms. $L_{\mathcal{D}}$ and $L_{\mathcal{K}}$ share the same parameters, i.e., the embedding vectors for words and their corresponding entities are the same. In the following subsections, we will introduce the text model to specify $L_{\mathcal{D}}$ and the knowledge model to specify $L_{\mathcal{K}}$.

3.1 Text Model

Similar to [10-11], we leverage the Skip-Gram model[5] as the text model. In Skip-Gram, the probability of observing the target word w_{O} given its context word w_{I} is modeled as $P(w_{\mathrm{O}}|w_{\mathrm{I}}) = \frac{\exp(\boldsymbol{w}'_{\mathrm{O}} \cdot \boldsymbol{w}_{\mathrm{I}})}{\sum_{w \in \mathcal{V}} \exp(\boldsymbol{w}' \cdot \boldsymbol{w}_{\mathrm{I}})}$, where $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{w}' \in \mathbb{R}^d$ denote the input and the output embedding vectors for word w respectively, $\mathcal{V}$ is the dictionary, and d is the dimension of the embedding.

Given the training corpus $\mathcal{D}$ consisting of $|\mathcal{D}|$ token words $\{p_1, \cdots, p_k, \cdots, p_{|\mathcal{D}|}\}$, loss $L_{\mathcal{D}}$ is specified by:

$$L_{\mathcal{D}} = \sum_{k=1}^{|\mathcal{D}|} \sum_{j \in \{-M, \cdots, M\}, j \neq 0} \log P(p_k | p_{k+j}),$$

where $2M$ is the size of the sliding window. As it is expensive to directly minimize $L_{\mathcal{D}}$ due to the denominator of $P(w_{\mathrm{O}}|w_{\mathrm{I}})$, we adopt the negative sampling strategy[5] to boost the computation efficiency.

3.2 Knowledge Model

The knowledge model in ProjectNet is based on an asymmetric low-rank projection that projects the original entity embedding vectors into a new semantic space. The projection is designed to be asymmetric in order to handle the heterogeneity between head and tail entities, and is designed to be low-rank in order to deal with non one-to-one relationships in the knowledge graphs.

3.2.1 Asymmetric Projection

As aforementioned, the head and tail entities in knowledge graphs are usually very different, from both semantic and statistical perspectives. Therefore, we argue that it is unreasonable to adopt the same projection to these two kinds of entities (as TransH[14] does). Instead, it would be better to adopt different projection matrices, denoted as $\boldsymbol{L}_r \in \mathbb{R}^{d \times d}$ and $\boldsymbol{R}_r \in \mathbb{R}^{d \times d}$ respectively, to the head and tail entities. Hence, given a triple (h, r, t), the original embedding vectors for h and t will be transformed to $\boldsymbol{h}'$ and $\boldsymbol{t}'$ as follows,

$$\boldsymbol{h}' = \boldsymbol{L}_r \boldsymbol{h}, \ \boldsymbol{t}' = \boldsymbol{R}_r \boldsymbol{t}. \tag{2}$$

Based on the transformed embeddings, we define a scoring function f_{dist} to reflect the confidence level that triple (h, r, t) is true:

$$f_{\mathrm{dist}}(h, r, t) = ||\boldsymbol{h}' + \boldsymbol{r} - \boldsymbol{t}'||_2^2 = ||\boldsymbol{L}_r \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{R}_r \boldsymbol{t}||_2^2. \tag{3}$$

Then we adopt a margin based ranking loss to distinguish the golden relationship triples from randomly corrupted triples:

$$L_{\mathcal{K}} = \sum_{(h,r,t)} \sum_{(h',r',t') \in N(h,r,t)} [\gamma - f_{\mathrm{dist}}(h', r', t') + f_{\mathrm{dist}}(h, r, t)]_+, \tag{4}$$

where $[x]_+ = \max(0, x)$, $\gamma > 0$ is the margin value, $N(h, r, t)$ is the set of all the corrupted triples built for triple (h, r, t), and $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$ will be specified in (5).

In our implementation, we used a trick that in (3) and (4), the sigmoid function is used to constrain every relationship vector $\boldsymbol{r}$ such that the scales of text loss and knowledge loss are better balanced in (1). To be more concrete, the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$ is applied to $\boldsymbol{r}$ element-wisely before calculating $f_{\mathrm{dist}}(h, r, t)$ and in such a case, the gradient of $f_{\mathrm{dist}}(h, r, t)$ with respect to r is: $\frac{\partial f_{\mathrm{dist}}(h,r,t)}{\partial \boldsymbol{r}} = 2(\boldsymbol{L}_r \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{R}_r \boldsymbol{r}) \circ \sigma(\boldsymbol{r}) \sigma(-\boldsymbol{r})$, where $\circ$ denotes element-wise product.

3.2.2 Low-Rank Projection

As mentioned in the introduction, many relationships in the knowledge graphs are non one-to-one. In this case, in order to achieve reasonable results, during the minimization of $L_{\mathcal{K}}$ defined above, it is necessary to constrain the projection matrices $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$ to be low-rank, which is described in the following proposition.

Proposition 1. *Once linear projections are imposed to head and tail entities, the necessary condition to overcome the non one-to-one mapping problem is that the projection matrices* $\boldsymbol{L}_r$ *and* $\boldsymbol{R}_r$ *should not be full-ranked.*

Proof. Consider the following least-square problem w.r.t. the optimization variable $\boldsymbol{h}$:

$$\min ||\boldsymbol{L}_r \boldsymbol{h} - \boldsymbol{c}||_2^2,$$

where $\boldsymbol{L}_r \boldsymbol{h} = \boldsymbol{h}'$ and we regard $\boldsymbol{c} = \boldsymbol{t}' - \boldsymbol{r}$ as a constant vector. It is easy to obtain that the optimal solution $\boldsymbol{h}^*$ satisfies the following linear system:

$$\boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{L}_r \boldsymbol{h}^* = \boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{c}.$$

To avoid the non one-to-one mapping problem, the above equation must have multiple solutions. Then it is necessary that $\boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{L}_r$ is a low-rank matrix. In addition, as $rank(\boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{L}_r) = rank(\boldsymbol{L}_r)$, the linear projection matrix $\boldsymbol{L}_r$ must not be full-rank either. The same conclusion holds for the projection matrix $\boldsymbol{R}_r$ for the tail entity. □

Given the above proposition, we use the following tricks to ensure that $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$ are low-rank matrices (whose ranks are m_L and m_R respectively, with $m_L < d$ and $m_R < d$):

$$\boldsymbol{L}_r = \sum_{i=1}^{m_L} \mu_r^{(i)} \boldsymbol{p}_r^{(i)} \boldsymbol{q}_r^{(i)\mathrm{T}}, \quad \boldsymbol{R}_r = \sum_{i=1}^{m_R} \zeta_r^{(i)} \boldsymbol{o}_r^{(i)} \boldsymbol{s}_r^{(i)\mathrm{T}}, \tag{5}$$

where $\mu_r^{(i)}$, $\zeta_r^{(i)}$ are scalars, and $\boldsymbol{p}_r^{(i)}$, $\boldsymbol{q}_r^{(i)}$, $\boldsymbol{o}_r^{(i)}$, $\boldsymbol{s}_r^{(i)}$ are all d-dimensional real vectors, the outer products of which constitute $(m_L + m_R)$ rank 1 matrices $\boldsymbol{p}_r^{(i)}\boldsymbol{q}_r^{(i)\mathrm{T}}$ and $\boldsymbol{o}_r^{(i)}\boldsymbol{s}_r^{(i)\mathrm{T}}$. For simplicity, we set the rank of all the left matrices $\boldsymbol{L}_r$ to be the same (m_L) and the rank of all the right matrices $\boldsymbol{R}_r$ to be the same (m_R). Please note that we can also specify different ranks for different relationships r and we provide the corresponding discussions to the experiments (i.e., Section 4).

3.3 Discussions

In this subsection, we discuss the connections of our proposed ProjectNet algorithm with a few previous studies and show that they are special cases of ProjectNet.

RNet. RNet refers to the knowledge models proposed in [10] and [11]. In fact, both models in the two studies try to minimize the same scoring function: $f_{\text{dist}}(h, r, t) = ||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_2^2$. Their only difference lies in how f_{dist} is minimized. In [11], a large margin ranking loss is adopted for the minimization of $f_{\text{dist}}(h, r, t)$, whereas in [10], an approximate softmax loss is used. It is clear that such a scoring function $f_{\text{dist}}(h, r, t)$ cannot handle either the non one-to-one relationships between entities or the heterogeneity between head and tail entities. To state it more formally, let us consider the relationship triples $(h_i, r, t), i \in 1, \cdots, N$, where all head entities h_i have the same relationship r with tail entity t. In the ideal case, if all $f_{\text{dist}}(h_i, r, t)$ are fully minimized, we will have $\boldsymbol{h}_i = \boldsymbol{t} - \boldsymbol{r}, \forall i \in 1, \cdots, N$, which implies that $\boldsymbol{h}_1 = \boldsymbol{h}_2 = \cdots = \boldsymbol{h}_N$. It means that all the embedding vectors for the head entities $\{h_i\}_{i=1}^N$ are the same, which is clearly unreasonable. We may encounter similar issues for one-to-many relationships $\{(h, r, t_j)\}_j$ and many-to-many relationships $\{(h_i, r, t_j)\}_{i,j}$.

Note that RNet corresponds to $\boldsymbol{L}_r = \boldsymbol{R}_r = \boldsymbol{I}_{d\times d}$ in (2) and since the identity matrix $\boldsymbol{I}_{d\times d}$ can be written in the form of (5), RNet can be regarded as a special case of ProjectNet.

TransH. TransH[14] is proposed to overcome the non one-to-one mapping problem. It first projects the entity embedding vectors $\boldsymbol{h}$ and $\boldsymbol{t}$ onto a hyperplane w.r.t. relationship r, and then the projected vectors $\boldsymbol{h}_\perp$ and $\boldsymbol{t}_\perp$ are used to define the scoring function f_{dist}. Specifically,

$$\begin{aligned} \boldsymbol{h}_\perp = \boldsymbol{h} - \boldsymbol{w}_r^{\mathrm{T}}\boldsymbol{h}\boldsymbol{w}_r,\ \boldsymbol{t}_\perp = \boldsymbol{t} - \boldsymbol{w}_r^{\mathrm{T}}\boldsymbol{t}\boldsymbol{w}_r, \\ f_{\text{dist}}(h, r, t) = ||\boldsymbol{h}_\perp + \boldsymbol{r} - \boldsymbol{t}_\perp||_2^2, \end{aligned} \tag{6}$$

where $\boldsymbol{w}_r \in \mathbb{R}^d$ is the normal vector of the hyperplane with unit length (i.e., $\boldsymbol{w}_r \cdot \boldsymbol{r} = 0$ and $||\boldsymbol{w}_r||_2 = 1$).

Our proposed ProjectNet model differs from TransH in two ways: 1) we adopt different projections to head and tail entities; 2) we adopt general projection matrices rather than a hyperplane based projection. Actually, TransH (6) can be regarded as a special case of ProjectNet (4), as shown below. Starting from (6), we have

$$\boldsymbol{h}_\perp = \boldsymbol{h} - \boldsymbol{w}_r^{\mathrm{T}}\boldsymbol{h}\boldsymbol{w}_r = \boldsymbol{h} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}}\boldsymbol{h} = (\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}})\boldsymbol{h}.$$

Hence, by substituting $\boldsymbol{L}_r = (\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}})$ in (2) and (3), we get TransH. We still need to check whether $\boldsymbol{L}_r = (\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}})$ can be written in the form of (5), i.e., the weighted sum of m_L rank-1 matrices, where $m_L < d$. We answer this question in the following two steps. 1) As $\boldsymbol{L}_r = (\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}})$ is an idempotent matrix (i.e., $\boldsymbol{L}_r\boldsymbol{L}_r = \boldsymbol{L}_r$) and $\boldsymbol{w}_r$ is a unit length vector, it holds that $rank(\boldsymbol{L}_r) = trace(\boldsymbol{L}_r) = d - 1$[24]. Therefore, $\boldsymbol{L}_r$ has $d - 1$ non-zero eigenvalues. Furthermore, by observing that the eigenvalues of $\boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}}$ are 0 and 1, we can conclude that $\boldsymbol{L}_r = (\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}})$ has 1 as one of its eigenvalues, corresponding to $d-1$ linearly independent eigenvectors, and 0 as its another eigenvalue, corresponding to one eigenvector. 2) Further considering that $\boldsymbol{L}_r$ is a real symmetric matrix, we can decompose $\boldsymbol{L}_r$ as $\boldsymbol{L}_r = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^{\mathrm{T}}$, where $\boldsymbol{U}_r = (\boldsymbol{u}_r^{(1)}, \boldsymbol{u}_r^{(2)}, \cdots, \boldsymbol{u}_r^{(d)}) \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\Sigma}_r = diag(1, 1, \cdots, 1, 0) \in \mathbb{R}^{d\times d}$, where $diag(x)$ means the diagonal matrix with vector $\boldsymbol{x}$ as its diagonal. The first $d-1$ columns $\{\boldsymbol{u}_r^{(i)}\}_{i=1}^{d-1}$ of $\boldsymbol{U}_r$ are all the unit-length eigenvectors of $\boldsymbol{L}_r$ corresponding to eigenvalue 1 and $\boldsymbol{\Sigma}_r$ stores all the eigenvalues of $\boldsymbol{L}_r$. Thus we can write $\boldsymbol{L}_r = \sum_{i=1}^{d-1}\boldsymbol{u}_r^{(i)}\boldsymbol{u}_r^{(i)\mathrm{T}}$.

The same procedure holds for the relation between $\boldsymbol{R}_r\boldsymbol{t}$ and $\boldsymbol{t}_\perp$. Then according to the above discussions, we can obtain the following proposition.

Proposition 2. *In the knowledge model of ProjectNet* (3) *and* (5), *letting* $m_L = m_R = d-1$, $\mu_r^{(i)} = \zeta_r^{(i)} = 1$ *and* $\boldsymbol{p}_r^{(i)} = \boldsymbol{q}_r^{(i)} = \boldsymbol{o}_r^{(i)} = \boldsymbol{s}_r^{(i)} = \boldsymbol{u}_r^{(i)}$, *where* $\boldsymbol{u}_r^{(i)}$ *is the i-th eigenvector of the matrix* $\boldsymbol{I} - \boldsymbol{w}_r\boldsymbol{w}_r^{\mathrm{T}}$ *with unit length,* $i = 1, 2, \cdots, d-1$, *we can obtain the TransH model* (6).

SE. SE[15] adopts the following scoring function:

$$f_{\text{dist}}(h, r, t) = ||\boldsymbol{L}_r\boldsymbol{h} - \boldsymbol{R}_r\boldsymbol{t}||_1.$$

SE looks very similar to our proposed knowledge model. However, there is a key difference: SE does not add the low-rank constraints to matrices $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$. In other words, they fix the rank of these two matrices to be full, while in our model, the rank of the matrices is a variable. Therefore our model is more general than SE and can handle the non one-to-one relationship when the rank is low while SE cannot, since its rank is always full. In this sense, we could also regard SE as a special case of our proposed ProjectNet model.

TransR. TransR[25] treats relationships and entities as different objects and thus separates their embeddings into different spaces,

$$f_{\text{dist}}(h, r, t) = ||\boldsymbol{M}_r\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{M}_r\boldsymbol{t}||_2^2.$$

Different from (4) and (5), the authors of [25] did not add the low-rank constraint to matrix $\boldsymbol{M}_r$ (or we say that it sets matrix $\boldsymbol{M}_r$ to be full-rank). In addition, TransR adopts the same transformation matrices to head and tail entities, by assuming that they are located in the same space. Therefore, the knowledge model in our ProjectNet algorithm is more general than TransR, and can include it as our special case. Moreover, in terms of parameter complexity, TransR uses total d^2 parameters in modeling projections for each relationship r, where ProjectNet uses $2(m_L+m_R)d+m_L+m_R$ parameters. When $2(m_L + m_R) > d$ (just the case in our experiments), ProjectNet owns more parameters than TransR and thus is more expressive.

4 Experiments

We conducted a set of experiments to verify the effectiveness of the ProjectNet model. The source code, together with dataset can be downloaded via the link https://github.com/ProjDLer/ProjectNet.

4.1 Experimental Setup

4.1.1 Training Data

For the free text corpus, we used a public snapshot of English Wikipedia named *enwik*9[1]. The corpus contains about 120 million word tokens. We removed digital words and words with frequency less than 5. Then we leveraged a knowledge graph $FB13$[13] to impose relationships onto those entities covered by *enwik*9. Since $FB13$ contains many entities whose names have multiple words, in *enwik*9, we merged these words into phrases and regarded both single words and phrases as embedding units in the dictionary. Finally the dictionary size is about 230k.

4.1.2 Baseline Methods

We considered the following algorithms as our baselines (we used the codes released by the authors of these studies for implementation):

1) *Skip-Gram* (*SG*): the original Skip-Gram model in *word2vec*, corresponding to $\alpha = 0$ in (1).

2) *RNet*: the joint embedding model in [10] and [11], which adopts the objective min $||\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}||_2^2$ in the knowledge model[2].

3) *Skip-Gram + TransH* ($SG+TransH$): the combination of Skip-Gram (for the text model) and TransH (for the knowledge model). According to the discussions in Subsection 3.3, this baseline is a special case of ProjectNet.

4.1.3 Parameter Settings

In our experiments, we set the embedding size to $d = 100$. Stochastic Gradient Descent (SGD) is used to train all the models. We set the initial learning rate to be 0.025 and linearly dropped it during the training process. For the knowledge model in ProjectNet, we initialized the projection matrices $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$ to be diagonal matrices with randomly assigned elements $0, 1$ (with m_L and m_R non-zero elements respectively). For m_L and m_R, we varied their values according to the set $\{10, 20, \cdots, 80, 90, 95, 100\}$. For all the joint embedding models, we varied the trade-off parameter α in (1) according to the set $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. The margin value is set to $\gamma = 1$.

We used two tasks to evaluate our algorithm and the baseline models: one is the analogical reasoning task and the other is the word similarity task. The corresponding experimental results are shown in the following two subsections.

4.2 Analogical Reasoning Task

The analogical reasoning task is a word relationship inference task proposed in [5]. It consists of several quadruple word questions $a : b, c : d$, in which the relationship between word a and word b is the same as that between c and d. For instance, $(a : b,\ c : d) = (Berlin : Germany, Paris : France)$ and the relationship r is *capital-countries*. The task aims to infer word d given words a, b, and c using their word embedding vectors. To be more concrete, the inferred word $\hat{d}$ is given by $\hat{d} = \arg\max_{w\in\mathcal{V}} \text{cosine}(\boldsymbol{b} - \boldsymbol{a} + \boldsymbol{c}, \boldsymbol{w})$. Once

[1] http://mattmahoney.net/dc/enwik9.zip, Feb. 2016.

[2] As aforementioned, the models in [10] and [11] differ only in the loss function (i.e., ranking loss vs approximate softmax loss). Hence, we unify these two models using the name RNet and report the better performance of the two loss functions.

$\hat{d} = d$, the result on this quadruple word question is right; otherwise, it is wrong.

The original analogical reasoning dataset given in [5] is too special in the sense that almost all the relationships in it are one-to-one mappings. For fair comparison, we constructed a dataset derived from real-world knowledge graph, that is $FB13$[3]. In addition, in Subsection 4.2.3, we still report the experimental results on Mikolov's original semantic/syntactic dataset[5] to make our experimental results more comprehensive.

To construct the test set for the analogical reasoning task, we randomly sampled 1% triples from $FB13$, and filtered them according to the dictionary of *enwik*9. This constructed dataset consists of about 20k questions belonging to seven non one-to-one relationships. The detailed statistics for this test dataset can be found in Table 2.

Then, we went through the remaining triples in $FB13$ and removed all those triples containing overlapped entities with the test data. In this way, we obtained a training set with roughly 76k triples, which has no overlap with the test set in either relationship triples or entities. The goal of doing so is to examine whether the free text corpus can act as a bridge between known and unknown entities, so as to verify the necessity of jointly embedding text and knowledge into the distributed representation space.

For ProjectNet, as we imposed $\boldsymbol{L}_r\boldsymbol{a} - \boldsymbol{R}_r\boldsymbol{b} = -\boldsymbol{r} = \boldsymbol{L}_r\boldsymbol{c} - \boldsymbol{R}_r\boldsymbol{d}$ instead of $\boldsymbol{a} - \boldsymbol{b} = \boldsymbol{c} - \boldsymbol{d}$, we took a two-step approach instead of directly using the original word vectors to perform the analogical reasoning task: 1) we chose an optimal relationship r^* that best describes the relationship between a and b, i.e., $r^* = \arg\min_r ||\boldsymbol{L}_r\boldsymbol{a} + \boldsymbol{r} - \boldsymbol{R}_r\boldsymbol{b}||_2^2$; 2) under r^*, we chose the answer word $\hat{d}$ according to $\hat{d} = \arg\min_{w\in\mathcal{V}} ||\boldsymbol{L}_{r^*}\boldsymbol{c} + r^* - \boldsymbol{R}_{r^*}\boldsymbol{w}||_2^2$. The same evaluation method was applied to SG+TransH as well.

The experimental results are listed in Table 2, from which we have the following observations.

- All the knowledge based models (RNet, SG +TransH, and ProjectNet) outperform the original SG model, indicating that the quality of word embedding can be improved by leveraging knowledge graphs.
- The two models that take non one-to-one mappings into consideration (i.e., SG+TransH and ProjecNet) are superior to RNet, showing the necessity of modeling the non one-to-one mappings into the loss functions.
- Among all the models, ProjectNet achieves the best performance in all the seven subtasks. For the overall accuracy, it achieves over 30% relative gain compared with SG+TransH. This well demonstrates the advantages of our proposed model.

4.2.1 Sensitivity to Different Ranks

The best performance of ProjectNet was obtained with $\alpha = 0.2$, $m_L = 50$, and $m_R = 90$. To show the influence of the ranks of the projection matrices, in Fig.1, we plotted two curves that reflect the performance of ProjectNet w.r.t. different rank values m_L and m_R: one curve corresponds to changing m_R while fixing $m_L = 50$, and the other corresponds to changing m_L while fixing $m_R = 90$. From the figure, we have the following observations. 1) The performance becomes bad when the rank is too low. This is because in this case the model expressiveness becomes poor due to a small number of free parameters in the projection matrices. 2) For the projection matrix for head entities, medium values of m_L correspond to the better performances (the red line), while for the projection matrix for tail entities, higher values of m_R lead to better performances (the green line). This result is consistent with the statistical information in Table 1: the degree of non one-to-one mappings for head entities is much higher

Table 2. Accuracy of Different Models on Analogical Reasoning Task

Relationship	#Question	SG (%)	RNet (%)	SG+TransH (%)	ProjectNet (%)
Cause_of_death	4 290	3.29	4.31	7.55	10.84
Nationality	870	14.60	14.14	14.82	17.47
Gender	650	67.08	59.38	75.54	84.62
Profession	6 320	3.73	5.78	8.66	13.42
Institution	4 556	1.54	3.03	4.92	6.72
Ethnicity	342	16.96	15.50	15.79	18.71
Religion	3 192	13.00	12.47	15.88	22.06
Total	**20 220**	**7.33**	**8.15**	**11.24**	**15.28**

[3] https://github.com/ProjDLer/ProjectNet/tree/master/dataset, Feb. 2016.

than that for tail entities, suggesting a lower rank of projection matrix for head entities.

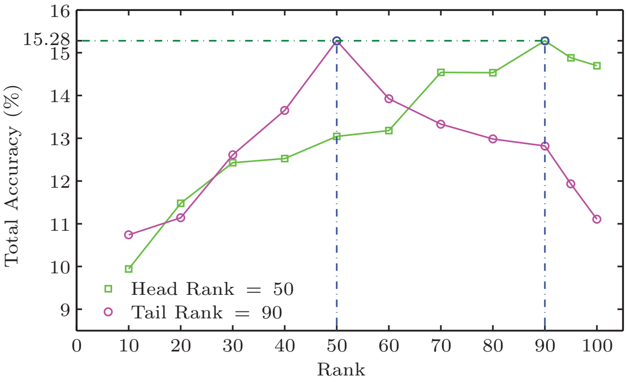

Fig.1. Accuracy w.r.t. different head and tail ranks. The red line records the accuracy varying with different head ranks when tail rank is fixed to 90. The green line records the accuracy varying with different tail ranks when head rank is fixed to 50.

Here we would like to give some practical suggestions on setting different rank values for the head project matrices $\boldsymbol{L}_r$ and the tail project matrices $\boldsymbol{R}_r$. First, the rank values of all the tail projection matrices $\boldsymbol{R}_r$ can be set near to full rank, given that *#Tail per Head* is only a little bit larger than 1. For example, Fig.1 shows that m_R values taking from $\{80, 90, 95, 100\}$ yield fairly good results. Second, for the head projection matrices $\boldsymbol{L}_r$, their rank values can be configured as an identical smaller value (e.g., $m_L = 50$ suggested by our experimental results). In addition, as stated in Subsection 3.2.2, a more reasonable way to configure rank values for head projection matrices is that for different relationships r, different rank values m_r are adopted. In practice, we found that setting $m_r = \max\ (10, \lfloor 100 - 10 \ln p_r \rfloor$ will generate good experimental result (i.e., total accuracy 14.86% for analogical reasoning task), where p_r is *#Tail per Head* values for relationship r.

4.2.2 Sensitivity to Different α Values

We report the parameter sensitivity of the trade-off parameter α in (1). Different accuracy scores on the new analogical reasoning dataset w.r.t. different α values are listed in Fig.2. From this figure, it can be observed that in a large range of α values (i.e., $\alpha \in (0.1, 0.5)$), the accuracy scores are fairly good (greater than 10%), which shows the robustness of ProjectNet with regard to the hyperparameter that balances text loss and knowledge loss.

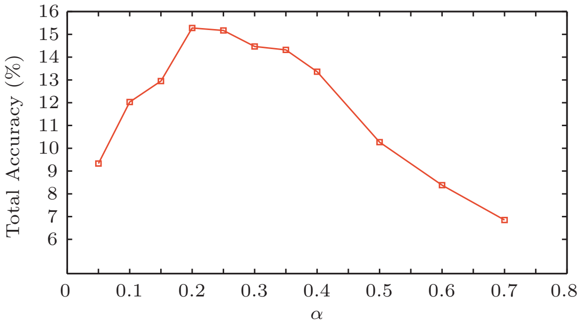

Fig.2. Accuracy w.r.t. different trade-off parameters α, when fixing $m_L = 50$ and $m_R = 90$.

4.2.3 Results on Miklov's Dataset

In Table 3, we report the experimental results on Miklov's semantic and syntactic dataset[5]. Since in this dataset, nearly all of the relationships are one-to-one mappings (such as *capital-countries*, *currency-countries* and *words-plural forms*), in our ProjectNet model, we imposed the head and tail projection matrices to be of full-rank, i.e., in (5), $m_L = m_R = d$.

Table 3. Accuracy of Different Models on Miklov's Analogical Dataset[5]

Task	SG (%)	RNet (%)	SG+TransH (%)	ProjectNet (%)
Semantic	25.06	26.91	25.13	26.62
Syntactic	36.49	39.37	35.89	38.76

In Table 3, the performance of Skip-Gram model and RNet is referenced from previous work[11]. We can observe that the best model on both semantic and syntactic dataset is the RNet model. Our ProjectNet performs comparatively with the best model. We conjectured on this dataset, ProjectNet is a little inferior because it has much more parameters in the two projection matrices $\boldsymbol{L}_r$ and $\boldsymbol{R}_r$, leading to more optimization difficulties. In addition, ProjectNet significantly outperforms the other baseline methods including original Skip-Gram and SG+TransH model.

4.3 Word Similarity Task

Word similarity is a task to investigate whether the similarity computed from word embedding vectors is consistent with human-labeled word similarity. We used three word-similarity tasks in our experiments, namely Word Similarity 353 (WS353)[26], SCWS[3] and Rare Word (RW)[19]. There are 353, 2 003, and 2 034 word pairs in these datasets respectively. From the word embedding vectors, we obtained the similarity scores (e.g., cosine similarity) for each word pair, based on which a ranked list is derived on the word pairs. Then the generated ranked list is compared with the ranked list produced by the ground-truth similarity scores assigned by human labelers. To evaluate the consistency between two ranking lists, we used Spearman's rank correlation (denoted as $\rho \in [-1, 1]$). Higher ρ corresponds to better word embedding vectors.

Table 4 summarizes the results. For ProjectNet and SG+TransH, the word embedding vectors were directly used to compute the similarity scores, which is different from the analogical reasoning task. This is because there is no explicit relationship available in the evaluation process. The best performances of ProjectNet on the three datasets were obtained with the parameters setting to $(m_L = 50, m_R = 95, \alpha = 0.05)$, $(m_L = 40, m_R = 90, \alpha = 0.01)$, and $(m_L = 60, m_R = 95, \alpha = 0.05)$ respectively. Table 4 reveals that ProjetNet achieves the best performance on all the datasets, which further indicates that ProjetNet produces higher quality word embedding vectors than the baseline methods.

Table 4. Spearman's Rank Correlation (ρ) on Three Word Similarity Datasets: WS353, SCWS, and RW

Task/Model	SG	RNet	SG+TransH	ProjectNet
WS353	0.647	0.661	0.666	**0.684**
SCWS	0.610	0.614	0.618	**0.630**
RW	0.179	0.184	0.187	**0.198**

Note: each ρ is reported as the average value of five repeated runs.

5 Conclusions

In this paper, we proposed a novel word embedding algorithm called ProjetNet, which leverages knowledge graphs to improve the quality of word embedding. In ProjetNet, we adopted different asymmetric low-rank projections to head and tail entities in an entity-relationship triple, and thus successfully maintained both non one-to-one mapping and heterogeneous head/tail entities properties of knowledge graph. Experimental results demonstrated that ProjetNet significantly outperforms previous embedding models.

For the future work, we plan to apply the proposed approach to fulfill knowledge mining tasks, such as triplet classification and link prediction[14]. In addition, we plan to use the word embedding vectors generated by ProjectNet in some real-world applications such as document classification and web search ranking.

References

[1] Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. *The Journal of Machine Learning Research*, 2003, 3: 1137-1155.

[2] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. the 25th International Conference on Machine Learning*, July 2008, pp.160-167.

[3] Huang E, Socher R, Manning C, Ng A. Improving word representations via global context and multiple word prototypes. In *Proc. the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, July 2012, Volume 1, pp.873-882.

[4] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv Preprint, arXiv:1301.3781, 2013. http://arxiv.org/pdf/1301.3781.pdf, Mar. 2016.

[5] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In *Proc. the 27th NIPS*, December 2013, pp.3111-3119.

[6] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation. In *Proc. the Empirical Methods in Natural Language Processing* (*EMNLP*), October 2014, pp.1532-1543.

[7] Miller G. WordNet: A lexical database for English. *Communications of the ACM*, 1995, 38(11): 39-41.

[8] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. the 2008 ACM SIGMOD International Conference on Management of Data*, June 2008, pp.1247-1250.

[9] Bian J, Gao B, Liu T. Knowledge-powered deep learning for word embedding. In *Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Sept. 2014, pp.132-148.

[10] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph and text jointly embedding. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), October 2014, pp.1591-1601.

[11] Xu C, Bai Y, Bian J, Gao B, Wang G, Liu X, Liu T. RC-NET: A general framework for incorporating knowledge into word representations. In *Proc. the 23rd ACM International Conference on Information and Knowledge Management*, November 2014, pp.1219-1228.

[12] Zheng Z, Si X, Li F, Chang E, Zhu X. Entity disambiguation with freebase. In *Proc. the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, December 2012, Volume 01, pp.82-89.

[13] Socher R, Chen D, Manning C, Ng A. Reasoning with neural tensor networks for knowledge base completion. In *Proc. the 27th Neural Information Processing Systems*, December 2013, pp.926-934.

[14] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In *Proc. the 28th AAAI Conference on Artificial Intelligence*, July 2014, pp.1112-1119.

[15] Bordes A, Weston J, Collobert R, Bengio Y. Learning structured embeddings of knowledge bases. In *Proc. the 25th AAAI Conference on Artificial Intelligence*, August 2011, pp.301-307.

[16] Turian J, Ratinov L, Bengio Y. Word representations: A simple and general method for semi-supervised learning. In *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, July 2010, pp.384-394.

[17] Botha J A, Blunsom P. Compositional morphology for word representations and language modelling. In *Proc. the 31st International Conference on Machine Learning*, June 2014, pp.1899-1907.

[18] Tian F, Dai H, Bian J, Gao B, Zhang R, Chen E, Liu T Y. A probabilistic model for learning multi-prototype word embeddings. In *Proc. the 25th International Conference on Computational Linguistics*, August 2014, pp.151-160.

[19] Luong M T, Socher R, Manning C. Better word representations with recursive neural networks for morphology. In *Proc. the 17th Conference on Computational Natural Language Learning*, August 2013, pp.104-113.

[20] Cui Q, Gao B, Bian J, Qiu S, Dai H, Liu T. KNET: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems* (*TOIS*), 2015, 34(1): Article No. 4.

[21] Yu M, Dredze M. Improving lexical embeddings with semantic knowledge. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics* (*Short Papers*), June 2014, pp.545-550.

[22] Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In *Proc. the 27th Neural Information Processing Systems*, December 2013, pp.2787-2795.

[23] Jenatton R, Le Roux N, Bordes A, Obozinski G R. A latent factor model for highly multi-relational data. In *Proc. the 26th Neural Information Processing Systems*, December 2012, pp.3167-3175.

[24] Horn R A, Johnson C R. Matrix Analysis. Cambridge University Press, 1990.

[25] Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In *Proc. the 29th AAAI Conference on Artificial Intelligence*, January 2015, pp.2181-2187.

[26] Finkelstein L, Gabrilovich E, Matias Y, Rivlin E, Solan Z, Wolfman G, Ruppin E. Placing search in context: The concept revisited. In *Proc. the 10th International Conference on World Wide Web*, May 2001, pp.406-414.

Fei Tian is now a Ph.D. candidate in the Department of Computer Science and Technology, University of Science and Technology of China, Hefei. He received his B.S. degree in computer science from University of Science and Technology of China, Hefei, in 2011. Since 2012, he has been a research intern with Machine Learning Group, Microsoft Research Asia in Beijing. His research interests mainly lie in neural networks and machine learning, especially deep learning methods for text mining and natural language processing.

Bin Gao is a lead researcher in Machine Learning Group, Microsoft Research Asia, Beijing. Prior to joining Microsoft, he got his Ph.D. degree in applied mathematics from Peking University, Beijing, in 2006, and his B.S. degree in computational mathematics from Shandong University, Jinan, in 2001. His research interests include machine learning, data mining, information retrieval, and computational advertising. He has authored two book chapters, 30 papers in top conferences and journals, and over 20 granted or pending patents. He co-authored the best student paper at SIGIR (2008). He serves as PC for SIGIR (2009, 2014), WWW (2011, 2013), and senior PC for CIKM (2011). He is a reviewer for TKDE, TIST, PRL, IRJ, etc. He is a tutorial speaker at WWW (2011) and SIGIR (2012). He is a workshop organizer at ICDM (2012), SIGIR (2013), KDD (2013), and ICML (2014). He is a member of ACM and IEEE.

En-Hong Chen received his Ph.D. degree from University of Science and Technology of China, Hefei, in 1996, his M.S. degree from Hefei University of Technology, Hefei, in 1992, and his B.S. degree from Anhui University, Hefei, in 1989, all in computer science. He is currently a professor and the vice dean of the School of Computer Science, the vice director of the National Engineering Laboratory for Speech and Language Information Processing of University of Science and Technology of China (USTC), and the winner of the National Science Fund for Distinguished Young Scholars of China. His research interests include data mining and machine learning, social network analysis and recommender systems. He has published lots of papers on refereed journals and conferences, including IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Mobile Computing, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, and SDM. He received the Best Application Paper Award on KDD-2008 and the Best Research Paper Award on ICDM-2011. He is a senior member of IEEE.

Tie-Yan Liu is a principle researcher and research manager at Microsoft Research Asia, Beijing. He got his Ph.D. degree in 2003 and Bachelor's degree in 1998, both in electronic engineering from Tsinghua University, Beijing. His research interests include machine learning, information retrieval, data mining, computational advertising, and algorithmic game theory. He is well known for his pioneer work on learning to rank for information retrieval. He has authored the first book in this area, and published tens of impactful papers on both algorithms and theorems of learning to rank. In addition, his paper on graph mining won the Best Student Paper Award of SIGIR (2008); his paper on video shot boundary detection won the Most Cited Paper Award of the Journal of Visual Communication and Image Representation (2004∼2006); and his work on Internet economics won the Research Break-Through Award of Microsoft Research Asia (2012). Tie-Yan is very active in serving the research community. He is a program committee co-chair of ACML (2015), WINE (2014), AIRS (2013), and RIAO (2010), a local co-chair of ICML 2014, a tutorial co-chair of SIGIR (2016) and WWW (2014), a doctorial consortium co-chair of WSDM (2015), a demo/exhibit co-chair of KDD (2012), and an area/track chair or senior program committee member of many conferences including KDD (2015), ACML (2014), SIGIR (2008∼2011), AIRS (2009∼2011), and WWW (2011, 2015). He is an associate editor of ACM Transactions on Information System (TOIS), and an editorial board member of Information Retrieval Journal (IRJ) and Foundations and Trends in Information Retrieval (FnTIR). He is a keynote speaker at ECML/PKDD (2014), CCIR (2011, 2014), CCML (2013), and PCM (2010), a tutorial speaker at SIGIR (2008, 2010, 2012), WWW (2008, 2009, 2011), and KDD (2012), and a plenary panelist at KDD (2011). He is a senior member of ACM and IEEE, as well as a senior member and distinguished speaker of CCF. He is currently an adjunct professor of the Language Technologies Institute (LTI) at Carnegie Mellon University, Nankai University, Sun Yat-sen University, and University of Science and Technology of China, and an honorary professor of University of Nottingham.