Elsevier Editorial System(tm) for Artificial Intelligence

Manuscript Draft

Manuscript Number:

Title: Probabilistic Belief Embedding for Large-scale Knowledge Population

Article Type: Full Length Article

Keywords: knowledge population; belief emebdding; entity inference; relation prediction; triplet.

Corresponding Author: Mr. Miao Fan,

Corresponding Author's Institution:

First Author: Miao Fan

Order of Authors: Miao Fan; Qiang Zhou, Ph.D.; Andrew Abel, Ph.D.; Fang Zheng, Ph.D.; Ralph Grishman, Ph.D.

Abstract: This paper contributes a novel embedding model which estimates the probability of each candidate belief <h,r,t,m> in a large-scale knowledge repository via simultaneously learning distributed representations for entities (h and t), relations (r), and the words in relation mentions (m). It facilitates knowledge population by means of simple vector operations to discover new beliefs. Given an imperfect belief, we can not only infer the missing entities, predict the unknown relations, but also tell the plausibility of the belief, just leveraging the learnt embeddings of remaining evidences. To demonstrate the scalability and the effectiveness of our model, we conduct experiments on several large-scale repositories which contain millions of beliefs from WordNet, Freebase and NELL, and compare it with other cutting-edge approaches via competing the performances assessed by the tasks of entity inference, relation prediction and triplet classification with their respective metrics. Extensive experimental results show that the proposed model outperforms the state-of-the-arts with significant improvements.

Suggested Reviewers:

Opposed Reviewers:

Probabilistic Belief Embedding for Large-scale Knowledge Population

Miao Fan[a], Qiang Zhou[a], Andrew Abel[a], Fang Zheng[a], Ralph Grishman[c]

[a] *Center for Speech and Language Technologies, Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China.*
[b] *Computing Science and Mathematics, School of Natural Sciences, Room 4B59, Cottrell Building, University of Stirling, Stirling FK9 4LA, U.K.*
[c] *Proteus Group, Computer Science Department, Courant Institute of Mathematical Sciences, New York University, 715 Broadway, 10003, NY, U.S.A.*

Abstract

This paper contributes a novel embedding model which estimates the probability of each candidate belief $\langle h, r, t, m\rangle$ in a large-scale knowledge repository via simultaneously learning distributed representations for entities (h and t), relations (r), and the words in relation mentions (m). It facilitates knowledge population by means of simple vector operations to discover new beliefs. Given an imperfect belief, we can not only infer the missing entities, predict the unknown relations, but also tell the plausibility of the belief, just leveraging the learnt embeddings of remaining evidences. To demonstrate the scalability and the effectiveness of our model, we conduct experiments on several large-scale repositories which contain millions of beliefs from WordNet, Freebase and NELL, and compare it with other cutting-edge approaches via competing the performances assessed by the tasks of *entity inference*, *relation prediction* and *triplet classification* with their respective metrics. Extensive experimental results show that the proposed model outperforms the state-of-the-arts with significant improvements.

Keywords: Knowledge population, belief emebdding, entity inference, relation prediction, triplet classification

URL: fanmiao.cslt.thu@gmail.com (Miao Fan), zq-lxd@mail.tsinghua.edu.cn (Qiang Zhou), aka@cs.stir.ac.uk (Andrew Abel), fzheng@tsinghua.edu.cn (Fang Zheng), grishman@cs.nyu.edu (Ralph Grishman)

1. Introduction

Information extraction [1, 2] has drawn much attention in recent years because of the explosive growth in the number of web pages. It is the study of extracting structured beliefs from unstructured online texts to populate knowledge bases. Thanks to the long-term efforts made by experts, crowdsouring and even machine learning techniques, several web-scale knowledge repositories, such as Wordnet[1], Freebase[2] and NELL[3], have been built. Among these knowledge repositories, WordNet [3] and Freebase [4, 5] follow the RDF format [6] that represents each belief as a triplet, i.e. $\langle head\ entity, relation, tail\ entity \rangle$, but NELL [7] goes a step further to extend each triplet with a *relation mention* which is a snatch of extracted free text to indicate the corresponding *relation*. Here we take a belief recorded in NELL as an example:

$\langle city : caroline, citylocatedinstate, stateorprovince : maryland, county\ and\ state\ of \rangle$,

in which *county and state of* is the mention between the head entity *city* : *caroline*, and the tail entity *stateorprovince* : *maryland*, to indicate the relation *citylocatedinstate*. In some cases, NELL also provides the $confidence$ of each belief automatically learnt by machines.

Although we have gathered colossal quantities of beliefs, state-of-the-art work [8] reports that our knowledge bases are far from complete. For instance, nearly 97% persons in Freebase have no records about their parents, whereas we human beings can still find the clue of their immediate family for most of the Freebase persons via searching on the web and looking up their Wiki. To populate the incomplete knowledge repositories assisted by computers, scientists either compete the performance of relation extraction between two named entities on manually annotated text datasets, such as ACE[4] and MUC[5], or look for effective approaches on improving the accuracy of link prediction within the

[1] http://wordnet.princeton.edu/
[2] https://www.freebase.com/
[3] http://rtw.ml.cmu.edu/rtw/
[4] http://www.itl.nist.gov/iad/mig/tests/ace/
[5] http://www.itl.nist.gov/iaui/894.02/relatedprojects/muc/

knowledge graphs constructed by the repositories without using extra free texts.

Recently, studies on text-based knowledge population have benefited a lot from a grateful paradigm called distantly supervised relation extraction (DSRE [9]) which bridges the gap between structured knowledge bases and unstructured free texts. It alleviates the labor of manual annotation by means of automatically aligning each triplet $\langle h, r, t \rangle$ from knowledge bases to the corresponding relation mention m in free texts. However the latest research [10] points out that DSRE still suffers from the problem of sparse and noisy features. Although Fan et al. fix the issue to some extent via leveraging the low-dimensional matrix factorization, the approach could not handle large-scale datasets as discussed in their academic article [10].

Fortunately, the knowledge embedding techniques [11, 12] enlighten us to encode the high-dimensional sparse features into low-dimensional distributed representations. A simple but effective model is TransE [13] which trains a vector representation for each entity and relation in large-scale knowledge bases without considering any text information. Even though Weston et al. [14], Wang et al. [15] and Fan et al. [16] broaden this field by adding word embeddings, there is still no comprehensive and elegant model that can integrate such large-scale heterogeneous resources to satisfy multiple subtasks of knowledge population including *entity inference*, *relation prediction* and *triplet classification*.

Therefore, we contribute a novel embedding model in this article, which estimates the probability of each candidate belief $\langle h, r, t, m \rangle$ in large-scale repositories. It breaks through the limitation of heterogeneous data, and establishes the connection between the structured knowledge graph and unstructured free texts. The distributed representations for entities (h and t), relations (r), as well as the words in relation mentions (m) are simultaneously learnt within the uniform framework of the probabilistic belief embedding (PBE) we propose. Then knowledge population can be facilitated by means of simple vector operations to discover new beliefs. Given an imperfect belief, we can not only infer the missing entities, predict the unknown relations, but tell the plausibility of the belief as well, just by means of the learnt vector representations of remain-

ing evidences. To prove the effectiveness and the scalability of PBE, we set up extensive experiments on multiple tasks, including *entity inference*, *relation prediction* and *triplet classification*, for knowledge population, and evaluate both our model and the cutting-edge approaches with appropriate metrics on several well-known large-scale repositories, such as WordNet, Freebase and NELL, which contain millions of beliefs. Elaborate results of comparison demonstrate that the proposed model outperforms the state-of-the-arts with significant improvements.

2. Related Work

We generally group the studies of knowledge population into three categories according to the diverse resources they use: text-based knowledge extraction, repository-based knowledge inference and hybrid-based knowledge population. As their individual names imply, the first research community extracts the relations between two recognized entities from text corpora, the second takes advantage of the link patterns within a knowledge graph to infer new triplets, and the third party suggests leveraging both the structure and unstructured information from both the text corpora and the knowledge graph. This paper contributes a novel embedding model for hybrid-based knowledge population, which stands on the boundary between the second and the third research communities, and we thus conduct experiments that mainly compare our approach with several state-of-the-arts mentioned in Section 2.2 and 2.3.

2.1. Text-based knowledge extraction

There exists a huge amount of unstructured electronic texts on the Web. To better understand these online data, we would like to create an intelligent system that can annotate all the data with the structure of our concerns. Normally, we concern more about the knowledge on relations between named entities. So far, off-the-shelf softwares have been available to help recognize entities in texts, and what we need to further study is to identify the semantic relations between

a pair of the annotated entities. But before we learn to extract the relations with supervised learning, we should annotate a portion of the data first, and the paradigms of annotation have two branches as follows.

2.1.1. Corpus-based extraction

Traditional approaches compete the performance of relation extraction on the public corpora, including ACE and MUC, which have been annotated by experts already. They choose different features extracted from the texts, like syntactic [17], kernel [18] or semantic parser features [19], and adopt discriminative classifiers, such as Perceptron and Support Vector Machine (SVM) to help predict the relations. There is a comprehensive survey [2] which shows more details about this branch of research.

2.1.2. Distantly supervised extraction

Mintz et al. [9] firstly adopt Freebase to *distantly supervise* Wikipedia to automatically generate annotated corpora. The basic alignment assumption is that if a pair of entities participate in a relation, *all sentences* that mention these entities in Wikipedia are labeled by the relation name from Freebase. Then we can extract a variety of textual features and learn a multi-class logistic regression classifier. Inspired by multi-instance learning, Riedel et al. [20] relax the strong assumption and replace *all sentences* with *at least one sentence*. Hoffmann et al. [21] point out that many entity pairs have more than one relation. They extend the multi-instance learning framework to the multi-label circumstance. Surdeanu et al. [22] propose a novel approach to multi-instance multi-label learning for relation extraction, which jointly models all the sentences in texts and all labels in knowledge bases for a given entity pair. The latest research [10] points out that the distant supervision paradigm still suffers from sparse and noisy features. Whereas Fan et al. [10] fix the issue by means of the low-dimensional matrix factorization, as discussed in their scholar, the approach could not handle large-scale datasets as well.

2.2. Repository-based knowledge inference

This research community aims at self-inferring new beliefs based on knowledge repositories without extra texts. It has two categories, namely graph-based inference models and embedding-based inference models. The principal differences between them are:

- *Symbolic representation v.s. Distributed representation*: Graph-based models regard the entities and relations as atomic elements, and represent them in a symbolic framework. In contrast, embedding-based models explore distributed representations via learning a low-dimensional continuous vector representation for each entity and relation.

- *Relation-specific v.s. Open-relation*: Graph-based models aim to induce rules or paths for a specific relation first, and then infer corresponding new beliefs. On the other hand, embedding-based models encode all relations into the same embedding space and conduct inference without any restriction on some specific relation.

2.2.1. Graph-based Inference

Graph-based inference models generally learn the representation for specific relations from the knowledge graph.

N-FOIL [23] learns first order Horn clause rules to infer new beliefs from the known ones. So far, it has helped to learn approximately 600 such rules. However, its ability to perform inference over large-scale knowledge repositories is currently still very limited.

PRA [24, 25, 26] is a data-driven random walk model which follows the paths from the head entity to the tail entity on the local graph structure to generate non-linear feature combinations representing the labeled relation, and uses logistic regression to select the significant features which contribute to classifying other entity pairs belonging to the given relation.

2.2.2. Embedding-based Inference

Embedding-based inference models usually design various scoring functions $f_r(h,t)$ to measure the plausibility of a triplet $\langle h,r,t\rangle$. The lower the dissimilarity of the scoring function $f_r(h,t)$ is, the higher the compatibility of the triplet will be.

Unstructured [13] is a naive model which exploits the occurrence information of the head and the tail entities without considering the relation between them. It defines a scoring function $||\mathbf{h}-\mathbf{t}||$, and this model obviously can not discriminate a pair of entities involving different relations. Therefore, *Unstructured* is commonly regarded as the baseline approach.

Distance Model (SE) [11] uses a pair of matrices (W_{rh}, W_{rt}), to characterize a relation r. The dissimilarity of a triplet is calculated by $||W_{rh}\mathbf{h}-W_{rt}\mathbf{t}||_1$. As pointed out by Socher et al. [27], the separating matrices W_{rh} and W_{rt} weaken the capability of capturing correlations between entities and corresponding relations, even though the model takes the relations into consideration.

Single Layer Model, proposed by Socher et al. [27] thus aims to alleviate the shortcomings of the *Distance Model* by means of the nonlinearity of a single layer neural network $g(W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$, in which $g=tanh$. The linear output layer then gives the scoring function: $\mathbf{u}_r^T g(W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$.

Bilinear Model [28, 29] is another model that tries to fix the issue of weak interaction between the head and tail entities caused by *Distance Model* with a relation-specific bilinear form: $f_r(h,t)=\mathbf{h}^T W_r \mathbf{t}$.

Neural Tensor Network (NTN) [27] designs a general scoring function: $f_r(h,t)=\mathbf{u}_r^T g(\mathbf{h}^T W_r \mathbf{t}+W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$, which combines the *Single Layer Model* and the *Bilinear Model*. This model is more expressive as the second-order correlations are also considered into the nonlinear transformation function, but the computational complexity is rather high.

TransE [13] is a canonical model different from all the other prior arts, which embeds relations into the same vector space of entities by regarding the relation r as a translation from h to t, i.e. $\mathbf{h}+\mathbf{r}=\mathbf{t}$. It works well on the beliefs

with ONE-TO-ONE mapping property but performs badly on multi-mapping beliefs. Given a series of facts associated with a ONE-TO-MANY relation r, e.g. $\langle h, r, t_1\rangle, \langle h, r, t_2\rangle, ..., \langle h, r, t_m\rangle$, *TransE* tends to represent the embeddings of entities on the MANY-side extreme close to each other which are hardly discriminated.

TransM [30] leverages the structure of the whole knowledge graph, and adjusts the learning rate which is specific to each relation based on the multiple mapping property of the relation.

TransH [31] is the state of the art approach as far as we know. It improves *TransE* by modeling a relation as a hyperplane, which makes it more flexible with regard to modeling beliefs with multi-mapping properties.

2.3. Hybrid-based knowledge population

Due to the diverse feature spaces between unstructured texts and structured beliefs, the challenge of connecting natural language and knowledge turns out to project the features into the same space and to merge them together for knowledge population. Fan et al. [16] have recently proposed that they can jointly learn the embedding representations for both relations and mentions to predict unknown relations between entities in NELL. However, the functionality of their latest method limits to the relation prediction task, as the correlations between entities and relations are ignored. Therefore, we look forward to a comprehensive model that can simultaneously consider entities, relations and even the relation mentions, and can integrate the heterogeneous resources to support multiple subtasks of knowledge population, such as *entity inference*, *relation prediction* and *triplet classification*.

3. Theory

The intuition of the subsequent theory is that: Not each belief we have learnt, i.e. $\langle head\ entity, relation, tail\ entity, mention\rangle$ abbreviated as $\langle h, r, t, m\rangle$, is perfect and complete enough [32]. We thus explore modeling the probability of

each belief, i.e. $Pr(h, r, t, m)$. It is assumed that $Pr(h, r, t, m)$ is collaboratively influenced by $Pr(h|r, t)$, $Pr(t|h, r)$ and $Pr(r|h, t, m)$, where $Pr(h|r, t)$ stands for the conditional probability of inferring the head entity h given the relation r and the tail entity t, $Pr(t|h, r)$ represents the conditional probability of inferring the tail entity t given the head entity h and the relation r, and $Pr(r|h, t, m)$ denotes the conditional probability of predicting the relation r between the head entity h and the tail entity t with the relation mention m extracted from free texts. Therefore, we define that the probability of a belief equals to the geometric mean of $Pr(h|r, t)Pr(r|h, t, m)Pr(t|h, r)$ as shown in the subsequent equation,

$$Pr(h, r, t, m) = \sqrt[3]{Pr(h|r, t)Pr(r|h, t, m)Pr(t|h, r)}. \tag{1}$$

Suppose that we have a certain repository Δ, such as WordNet, which contains thousands of beliefs validated by experts. The learning object is intuitively set to maximize $\mathcal{L}_{max}$, where

$$\mathcal{L}_{max} = \prod_{\langle h,r,t,m\rangle \in \Delta} Pr(h, r, t, m). \tag{2}$$

In most cases, we can automatically build much larger but imperfect knowledge bases as well via crowdsouring (Freebase) and machine learning techniques (NELL). However, each belief of NELL has a confidence-weighted score c to indicate its plausibility to some extent. Therefore, we propose an alternative goal which aims at minimizing $\mathcal{L}_{min}$, in which

$$\mathcal{L}_{min} = \prod_{\langle h,r,t,m,c\rangle \in \Delta} \frac{1}{2}[Pr(h, r, t, m) - c]^2. \tag{3}$$

To facilitate the optimization progress, we prefer using the loglikelihood of $\mathcal{L}_{max}$ and $\mathcal{L}_{min}$, and the learning targets can be further processed as follows,

$$\begin{aligned} & \arg\max_{h,r,t,m} \quad \log \mathcal{L}_{max} \\ = \ & \arg\max_{h,r,t,m} \sum_{\langle h,r,t,m\rangle \in \Delta} \log Pr(h, r, t, m) \\ = \ & \arg\max_{h,r,t,m} \sum_{\langle h,r,t,m\rangle \in \Delta} \frac{1}{3} \left[\log Pr(h|r, t) + \log Pr(r|h, t, m) + \log Pr(t|h, r)\right]; \end{aligned} \tag{4}$$

$$
\begin{aligned}
& \arg\min_{h,r,t,m} \quad \log \mathcal{L}_{min} \\
= \; & \arg\min_{h,r,t,m} \sum_{\langle h,r,t,m,c\rangle \in \Delta} \frac{1}{2}[\log Pr(h,r,t,m) - \log c]^2 \\
= \; & \arg\min_{h,r,t,m} \sum_{\langle h,r,t,m,c\rangle \in \Delta} \frac{1}{2}\{\frac{1}{3}\ [\log Pr(h|r,t) + \log Pr(r|h,t,m) + \log Pr(t|h,r)] - \log c\}^2.
\end{aligned}
\tag{5}
$$

The advantage of the conversions above is that we can separate the factors out, compared with Equation (1), and what left for us is to figure out the approaches on modeling $Pr(h|r,t)$, $Pr(r|h,t,m)$ and $Pr(t|h,r)$.

$Pr(r|h,t,m)$ leverages the evidences from two different resources to predict the relation. If the concurrence of the two entities (h and t) in knowledge bases is independent of the appearance of the relation mention m from free texts, we can factorize $Pr(r|h,t,m)$ as shown by Equation (6):

$$
Pr(r|h,t,m) = Pr(r|h,t)Pr(r|m). \tag{6}
$$

Then we need to consider formulating $Pr(h|r,t)$, $Pr(r|h,t)$, $Pr(t|h,r)$ and $Pr(r|m)$, respectively.

Figure 1(a) illustrates the traditional way of recording knowledge as triplets. The triplets $\langle h,r,t\rangle$ can construct a knowledge graph in which entities (h and t) are nodes and the relation (r) between them is a directed edge from the head entity (h) to the tail entity (t). This kind of symbolic representation, whilst being very efficient for storing, is not flexible enough to statistical learning approaches [11]. But once we project each elements, including entities and relations in the knowledge repository, into the same embedding space, we can use

$$
\mathscr{D}(h,r,t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}|| + \alpha, \tag{7}
$$

a simple vector operation to measure the distance between $\mathbf{h}+\mathbf{r}$ and $\mathbf{t}$, in which h, r and t are encoded in d dimensional vectors, and α is the bias parameter. To estimate the conditional probability of appearing t given h and r, i.e. $Pr(t|h,r)$,

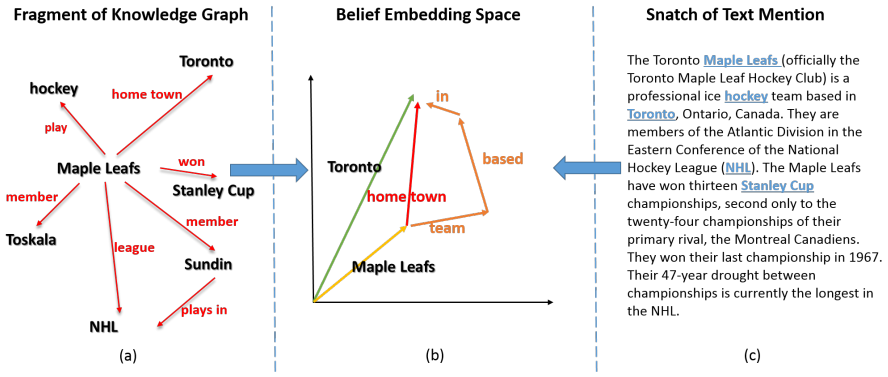

Figure 1: The whole framework of belief embedding. (a) shows a fragment of knowledge graph; (c) is a snatch of Wiki which describes the knowledge graph of (a); (b) illustrates how the belief $\langle Maple\ Leafs, home\ town, Toronto, team\ based\ in \rangle$ is projected into the same embedding space.

however, we need to adopt the softmax function[6] as follows,

$$Pr(t|h,r) = \frac{\exp^{\mathscr{D}(h,r,t)}}{\sum_{t' \in E_t} \exp^{\mathscr{D}(h,r,t')}}, \tag{8}$$

where E_t is the set of tail entities which contains all possible entities t' appearing in the tail position. Similarly, we can regard $Pr(h|r,t)$ and $Pr(r|h,t)$ as

$$Pr(h|r,t) = \frac{\exp^{\mathscr{D}(h,r,t)}}{\sum_{h' \in E_h} \exp^{\mathscr{D}(h',r,t)}} \tag{9}$$

and

$$Pr(r|h,t) = \frac{\exp^{\mathscr{D}(h,r,t)}}{\sum_{r' \in R} \exp^{\mathscr{D}(h,r',t)}}, \tag{10}$$

in which E_h is the set of head entities which contains all possible entities h' appearing in the head position, and R is the set of all candidate relations r'.

One the other hand, Figure 1(c) shows that free texts can provide fruitful contexts between two recognized entities, but the one-hot[7] feature space is rather high and sparse. Therefore, we can also project each words in relation

[6] http://en.wikipedia.org/wiki/Softmax_function

[7] http://en.wikipedia.org/wiki/One-hot

mentions into the same embedding space of entities and relations. To measure the similarity between the mention m and the corresponding relation r, we adopt inner product of their embeddings as shown by Equation (11),

$$\mathscr{F}(r, m) = \mathbf{W}^T \phi(m)\mathbf{r} + \beta, \tag{11}$$

where $\mathbf{W}$ is the matrix of $\mathbb{R}^{n_v \times d}$ containing n_v vocabularies with d dimensional embeddings, $\phi(m)$ is the sparse one-hot representation of the mention indicating absence or presence of words, $r \in \mathbb{R}^d$ is the embedding of relation r, and β is the bias parameter. Similar to Equation (8), (9) and (10), the conditional probability of predicting relation r given mention m, i.e. $Pr(r|m)$ can be defined as,

$$Pr(r|m) = \frac{\exp^{\mathscr{F}(r,m)}}{\sum_{r' \in R} \exp^{\mathscr{F}(r',m)}}. \tag{12}$$

Above all, we can finally model the probability of a belief via jointly embedding the entities, relations and even the words in mentions as demonstrated by Figure 1(b).

4. Algorithm

To search for the optimal solutions of Equation (4) and (5), we can use *Stochastic Gradient Descent*[8] (SGD) to update the embeddings of entities, relations and words of mentions in iterative fashion. However, it costs a lot to compute the normalization terms in $Pr(h|r,t)$, $Pr(r|h,t)$, $Pr(t|h,r)$ and $Pr(r|m)$ according to the their definitions made by Equation (8), (9), (10) and (12) respectively. For instance, if we directly calculate the value of $Pr(h|r,t)$ for just one belief, tens of thousands $\exp^{\mathscr{D}(h',r,t)}$ need to be re-valued, as there are tens of thousands candidate entities h' in E_h.

Enlightened by the work of Mikolov et al. [33], we have found an efficient approach that adopts negative sampling technique to approximate the conditional probability functions, i.e. Equation (8), (9), (10) and (12), by being transformed

[8] http://en.wikipedia.org/wiki/Stochastic_gradient_descent

to binary classification problems shown as the subsequent equations respectively,

$$\log Pr(h|r,t) \approx \log Pr(1|h,r,t) + \sum_{i=1}^{k} \mathbb{E}_{h'_i Pr(h' \in E_h)} \log Pr(0|h'_i,r,t), \tag{13}$$

$$\log Pr(t|h,r) \approx \log Pr(1|h,r,t) + \sum_{i=1}^{k} \mathbb{E}_{t'_i Pr(t' \in E_t)} \log Pr(0|h,r,t'_i), \tag{14}$$

$$\log Pr(r|h,t) \approx \log Pr(1|h,r,t) + \sum_{i=1}^{k} \mathbb{E}_{r'_i Pr(r' \in R)} \log Pr(0|h,r'_i,t), \tag{15}$$

$$\log Pr(r|m) \approx \log Pr(1|r,m) + \sum_{i=1}^{k} \mathbb{E}_{r'_i Pr(r' \in R)} \log Pr(0|r'_i,m), \tag{16}$$

where we sample k negative beliefs and discriminate them from the positive case. For the simple binary classification problems mentioned above, we choose the logistic function with the offset ϵ shown in Equation (17) to estimate the probability that the given triplet $\langle h,r,t \rangle$ is correct:

$$Pr(1|h,r,t) = \frac{1}{1+\exp^{-\mathscr{D}(h,r,t)}} + \epsilon, \tag{17}$$

and with the offset η shown in Equation (18) to tell the probability of the occurrence of r and m:

$$Pr(1|r,m) = \frac{1}{1+\exp^{-\mathscr{F}(r,m)}} + \eta. \tag{18}$$

We also display the framework of *PBE* learning algorithm written in pseudocode as shown by Algorithm 1.

5. Experiment

Besides its access to the efficient SGD algorithm, the learnt embeddings by *PBE* can contribute more effectiveness on multiple subtasks of knowledge population, such as entity inference, relation prediction, and triplet classification.

- *Entity inference*: Given a wrecked triplet, like $\langle h,r,? \rangle$ or $\langle ?,r,t \rangle$, the subtask works on inferring the missing entities to complete the triplet.

ALGORITHM 1 : The Learning Algorithm of PBE

Input:

Training set $\Delta = \{(h, r, t, m, c)\}$, entity set E, relation set R, vocabulary set V of relation mentions; dimension of embeddings d, number of negative samples k, learning rate γ, maximum epoches n; the bias α and β, the offset ϵ and η.

foreach $\mathbf{e} \in E$ **do**
  $\mathbf{e} :=$ Uniform$(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
end foreach
foreach $\mathbf{r} \in R$ **do**
  $\mathbf{r} :=$ Uniform$(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
end foreach
foreach $\mathbf{v} \in V$ **do**
  $\mathbf{v} :=$ Uniform$(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$
end foreach
$i := 0$
while $i < n$ **do**
  foreach $\langle h, r, t, m, c\rangle \in \Delta$ **do**
    foreach $j \in$ range(k) **do**
      Negative sampling: $\langle h'_j, r, t, m\rangle \in \Delta'_h$
      /*Δ'_h is the set of k negative beliefs replacing h*/
      Negative sampling: $\langle h, r'_j, t, m\rangle \in \Delta'_r$
      Negative sampling: $\langle h, r, t'_j, m\rangle \in \Delta'_t$
    end foreach
    Gradient ascent: $\sum_{h,r,t,h',r',t',v \in m} \nabla \log Pr(h, r, t, m)$ according to Equation (4)
    OR
    Gradient descent: $\sum_{h,r,t,h',r',t',v \in m} \nabla [\log Pr(h, r, t, m) - \log c]^2$ according to Equation (5)
    /*Updating embeddings of $\langle h, r, t, m\rangle \in \Delta$; $\langle h', r, t, m\rangle \in \Delta'_h$; $\langle h, r', t, m\rangle \in \Delta'_r$; $\langle h, r, t', m\rangle \in \Delta'_t$ with γ and the batch gradients derived from Equation (13), (14), (15) and (16).*/
  end foreach
  i++
end while

Output:

All the embeddings of h, t, r and v, where $h, t \in E$, $r \in R$ and $v \in V$.

- *Relation prediction*: Given a pair of entities and the text mentions indicating the semantic relations between them, i.e. $\langle h, ?, t, m \rangle$, this subtask predicts the best relations of the two entities.

- *Triplet classification*: It tells whether a completed triplet is correct or not ($\langle h, r, t \rangle ? \ 1 : 0$).

5.1. Entity inference

One of the benefits of knowledge embedding is that simple vector operations can apply to entity inference which contributes to knowledge graph completion. For example, if we would like to know which entity $h \in E_h$ is the exact head entity given the relation r and the entity t, we just need to compute the $\arg\max_{h \in E_h} Pr(h|r,t)$, with the help of the entity and relation embeddings. In the meanwhile, $\arg\max_{t \in E_t} Pr(t|h,r)$ will help us to find the best tail entity given the head entity h and the relation r.

5.1.1. Dataset

To demonstrate the wide adaptability of our approach, we prepare four datasets, i.e. **NELL-50K**, **WN-100K**, **FB-500K** and **NELL-1M** from the repositories of NELL [34], WordNet [3] and Freebase [4, 5], with varies scales as shown by Table 1. The NELL [35] designed and maintained by Carnegie Mellon University is an outstanding system which runs 24 hours/day and never stops learning the beliefs on the Web. Since the starting date of January 2010, it has acquired a knowledge repository with over 80 million confidence-weighted beliefs so far. The dataset **NELL-50K** we adopt, contains about fifty thousand training beliefs from NELL, and each belief has been validated to be true. We also extract a much larger one (**NELL-1M**) with one million training examples from NELL, where each belief is automatically learnt by machine and weighted ranging (0.5, 1.0). **WN-100K** is made by experts from WordNet, and owns only 11 kinds of relations but much more entities. Therefore, it is a sparse repository

in which fewer entities have connections. The last dataset (**FB-500K**[9]) we use was released by Bordes et al. [13]. It is a large but dense, crowdsourcing dataset extracted from Freebase, in which almost every two entities have connections, and each belief is a triplet without a confidence score.

DATASET	NELL-50K	WN-100K	FB-500K	NELL-1M
#(ENTITIES)	29,904	38,696	14,951	82,691
#(RELATIONS)	233	11	1,345	218
#(TRAINING EX.)	57,356	112,581	483,142	1,000,000
#(VALIDATING EX.)	10,710	5,218	50,000	24,864
#(TESTING EX.)	10,711	21,088	59,071	24,863

Table 1: Statistics of the datasets used for the entity inference task.

Table 1 shows the statistics of these four datasets. The statistical characteristic of these datasets are different, which may lead to the variety of tuning parameters.

5.1.2. Metric

For each testing belief, all the other entities that appear in the training set take turns to replace the head entity. Then we get a bunch of candidate triplets. The plausibility of each candidate triplet is firstly computed by various scoring functions, such as $Pr(h|r,t)$ in *PBE*, and then sorted in ascending order. Finally, we locate the ground-truth triplet and record its rank. This whole procedure runs in the same way when replacing the tail entity, so that we can gain the mean results. We use two metrics, i.e. *Mean Rank* and *Mean Hit@10* (the proportion of ground truth triplets that rank in Top 10), to measure the performance. However, the results measured by those metrics are relatively *raw*, as the procedure above tends to generate false negative triplets. In other

[9]We change the original name of the dataset (**FB15K**), so as to follow the naming conventions in our paper. Related studies on this dataset can be looked up from the website `https://www.hds.utc.fr/everest/doku.php?id=en:transe`

words, some of the candidate triplets rank rather higher than the ground truth triplet just because they also appear in the training set. We thus filter out those triplets to report more reasonable results.

DATASET	**NELL-50K**			
METRIC	*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*
TransE [13]	2,436 / 29,904	2,426 / 29,904	18.9%	19.6%
TransM [30]	2,296 / 29,904	2,285 / 29,904	20.5%	21.3%
TransH [31]	2,185 / 29,904	2,072 / 29,904	21.6%	**28.8%**
PBE	**2,078** / 29,904	**1,996** / 29,904	**22.5%**	26.4%

Table 2: Entity inference results on the **NELL-50K** dataset. We compared our proposed PBE with the state-of-the-art method TransH and other prior arts mentioned in Section 2.2.

DATASET	**WN-100K**			
METRIC	*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*
TransE [13]	10,623 / 38,696	10,575 / 38,696	3.8%	4.1%
TransM [30]	14,586 / 38,696	13,276 / 38,696	1.8%	2.0%
TransH [31]	12,542 / 38,696	12,463 / 38,696	2.3%	2.6%
PBE	**8,462** / 38,696	**8,409** / 38,696	**9.0%**	**10.1**%

Table 3: Entity inference results on the **WN-100K** dataset. We compared our proposed PBE with the state-of-the-art method TransH and other prior arts mentioned in Section 2.2.

5.1.3. Performance

We compare *PBE* with the state-of-the-art *TransH*, *TransM*, *TransE* mentioned in Section 2.2 via evaluating their performances on **NELL-50K**, **WN-100K**, **FB-500K**, and **NELL-1M** datasets. We tune the parameters of each previous model based on the validation set, and select the combination of parameters which leads to the best performance. To make responsible compar-

DATASET	FB-500K			
METRIC	*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*
TransE [13]	243 / 14,951	125 / 14,951	34.9%	47.1%
TransM [30]	196 / 14,951	93 / 14,951	44.6%	55.2%
TransH [31]	211 / 14,951	84 / 14,951	42.5%	58.5%
PBE	**165** / 14,951	**61** / 14,951	**50.5%**	**64.6%**

Table 4: Entity inference results on the **FB-500K** dataset. We compared our proposed PBE with the state-of-the-art method TransH and other prior arts mentioned in Section 2.2.

DATASET	NELL-1M			
METRIC	*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*
TransE [13]	29,059 / 82,691	29,052 / 82,691	6.5%	6.6%
TransM [30]	28,435 / 82,691	28,129 / 82,691	5.4%	5.5%
TransH [31]	27,455 / 82,691	26,980 / 82,691	7.8%	8.7%
PBE	**7,528** / 82,691	**7,485** / 82,691	**8.7%**	**9.0%**

Table 5: Entity inference results on the **NELL-1M** dataset. We compared our proposed PBE with the state-of-the-art method TransH and other prior arts mentioned in Section 2.2.

isons between *PBE* and the state-of-the-art approach *TransH*, we request its authors [31] to re-evaluate their system with all the four datasets and to report the best results. For *PBE*, we tried several combinations of parameters: $d = \{20, 50, 100\}$, $\gamma = \{0.1, 0.05, 0.01, 0.005\}$, and $norm = \{L_1, L_2\}$, and finally chose $d = 50$, $\gamma = 0.01$, $norm = L_2$ for **NELL-50K** and **WN-100K** datasets, and $d = 100$, $\gamma = 0.01$, $norm = L_2$ for **FB-500K** and **NELL-1M** datasets to conduct further experiments.

Table 2, 3, 4 and 5 demonstrate that *PBE* outperforms all the state-of-the-arts, including *TransE* [13], *TransM* [30] and *TransH* [31], and achieves significant improvements on all datasets. Overall, The *relative increments* performed by *PBE* compared with the best results of prior arts under all metrics are as subsequence,

- **NELL-50K**: {*Mean Rank Raw*: 4.9% ⇑, *Hit@10 Raw*: 4.2% ⇑, *Mean Rank Filter*: 3.7% ⇑, *Hit@10 Filter*: 8.3% ⇓};
- **WN-100K**: {*Mean Rank Raw*: 20.3% ⇑, *Hit@10 Raw*: 136.8% ⇑, *Mean Rank Filter*: 20.5% ⇑, *Hit@10 Filter*: 146.3% ⇑};
- **FB-500K**: {*Mean Rank Raw*: 15.8% ⇑, *Hit@10 Raw*: 27.3% ⇑, *Mean Rank Filter*: 13.3% ⇑, *Hit@10 Filter*: 10.4% ⇑};
- **NELL-1M**: {*Mean Rank Raw*: 72.5% ⇑, *Hit@10 Raw*: 11.5% ⇑, *Mean Rank Filter*: 72.2% ⇑, *Hit@10 Filter*: 3.4% ⇑}

5.2. Relation prediction

The scenario of this subtask is that: given a pair of entities and a short text/mention indicating the correct relations, we compute the $\arg\max_{r \in R} Pr(r|h,t)Pr(r|m)$ to predict the best relations.

5.2.1. Dataset

We continue using the datasets mentioned in Section 5.1 to compare the performances among all the competing methods. But, as the words in relation

mentions are additionally concerned the in this subtask, we also show the vocabulary size of relation mentions in each dataset in Table 6 as follows, except for **WN-100K** and **FB-500K** which only contain triplets as beliefs, and the sizes of their vocabulary are null.

DATASET	NELL-50K	WN-100K	FB-500K	NELL-1M
#(ENTITIES)	29,904	38,696	14,951	82,691
#(RELATIONS)	233	11	1,345	218
#(VOCABULARY)	8,948	-	-	12,354
#(TRAINING EX.)	57,356	112,581	483,142	1,000,000
#(VALIDATING EX.)	10,710	5,218	50,000	24,864
#(TESTING EX.)	10,711	21,088	59,071	24,863

Table 6: Statistics of the datasets used for the relation prediction task.

5.2.2. Metric

We compare the performances between our models and other state-of-the-art approaches mentioned in Section 2.2 and 2.3, including *TransE* [13], *TransM* [30], *TransH* [31] and *JRME* [16], with the metrics as follows,

- *Average Rank*: Each candidate relation will gain a score calculated by Equation (7). We sort them in ascent order and compare with the corresponding ground-truth belief. For each belief in the testing set, we get the rank of the correct relation. The average rank is an aggregative indicator, to some extent, to judge the overall performance on relation extraction of an approach.

- *Hit@10*: Besides the average rank, scientists from the industrials concern more about the accuracy of extraction when selecting Top10 relations. This metric shows the proportion of beliefs that we predict the correct relation ranked in Top10.

- *Hit@1*: It is a more strict metric that can be referred by automatic system, since it demonstrates the accuracy when just picking the first predicted relation in the sorted list.

DATASET	**NELL-50K**		
METRIC	**AVG. R.**	**HIT@10**	**HIT@1**
TransE [13]	131.8 / 233	16.3%	3.0%
TransM [30]	70.2 / 233	18.9%	4.3%
TransH [31]	46.3 / 233	20.0%	5.1%
JRME [16]	6.2 / 233	87.8%	60.2%
PBE	**2.5** / 233	**96.6%**	**78.3%**

Table 7: Performance of relation prediction on TransE, TransM, TransH, JRME and PBE evaluated by the metrics of Average Rank, Hit@10 and Hit@1 with **NELL-50K** dataset.

DATASET	**WN-100K**		
METRIC	**AVG. R.**	**HIT@10**	**HIT@1**
TransE [13]	3.8 / 11	98.3%	15.1%
TransM [30]	4.6 / 11	97.5%	14.8%
TransH [31]	3.4 / 11	99.0%	19.3%
JRME [16]	3.9 / 11	99.0%	15.9%
PBE	**2.0** / 11	**99.1%**	**72.6%**

Table 8: Performance of relation prediction on TransE, TransM, TransH, JRME and PBE evaluated by the metrics of Average Rank, Hit@10 and Hit@1 with **WN-100K** dataset.

5.2.3. Performance

Table 7, 8, 9 and 10 illustrate the results of experiments on relaton prediction with **NELL-50K**, **WN-100K**, **FB-500K** and **NELL-1M** datasets, respectively. All of them show that *PBE* performs best compared with all the latest approaches including the state-of-the-art *JRME* [16]. The relative increments are

DATASET	FB-500K		
METRIC	**AVG. R.**	**HIT@10**	**HIT@1**
TransE [13]	762.7 / 1,345	7.3%	1.9%
TransM [30]	402.3 / 1,345	13.4%	3.2%
TransH [31]	79.5 / 1,345	39.2%	15.6%
JRME [16]	60.9 / 1,345	27.4%	7.2%
PBE	**2.6** / 1,345	**97.3%**	**66.7%**

Table 9: Performance of relation prediction on TransE, TransM, TransH, JRME and PBE evaluated by the metrics of Average Rank, Hit@10 and Hit@1 with **FB-500K** dataset.

DATASET	NELL-1M		
METRIC	**AVG. R.**	**HIT@10**	**HIT@1**
TransE [13]	70.4 / 218	5.4%	0.4%
TransM [30]	65.5 / 218	18.7%	3.4%
TransH [31]	62.9 / 218	26.8%	5.8%
JRME [16]	7.0 / 218	89.0%	54.5%
PBE	**5.8** / 218	**92.1%**	**65.0%**

Table 10: Performance of relation prediction on TransE, TransM, TransH, JRME and PBE evaluated by the metrics of Average Rank, Hit@10 and Hit@1 with **NELL-1M** dataset.

- **NELL-50K**: {*Mean Rank*: 59.7% ⇑, *Hit@10*: 10.0% ⇑, *Hit@1*: 30.0% ⇑};
- **WN-100K**: { *Mean Rank*: 41.1% ⇑, *Hit@10*: 0.1% ⇑, *Hit@1*: 276.2% ⇑ };
- **FB-500K**: { *Mean Rank*: 95.7% ⇑, *Hit@10*: 148.2% ⇑, *Hit@1*: 327.6% ⇑ };
- **NELL-1M**: { *Mean Rank*: 20.6% ⇑, *Hit@10*: 3.5% ⇑, *Hit@1*: 19.3% ⇑ }.

Moreover, the leading results of *PBE* and *JRME* on **NELL** datasets also inspire us that text mentions can contribute a lot on predicting the correct relations.

5.3. Triplet classification

Triplet classification is another inference related task proposed by Socher et al. [27] which focuses on searching a relation-specific threshold σ_r to identify whether a triplet $\langle h, r, t\rangle$ is plausible. If the probability of a testing triplet (h, r, t) computed by $Pr(h|r,t)Pr(r|h,t)Pr(t|h,r)$ is below the relation-specific threshold σ_r, it is predicted as positive, otherwise negative.

5.3.1. Dataset

It is emphasized that the head or the tail entity can be randomly replaced with another one to produce a negative training example, but in order to build much tough validation and testing datasets, we constrain that the picked entity should once appear at the same position. For example, *(Pablo Picaso, nationality, U.S.)* is a potential negative example rather than the obvious nonsense *(Pablo Picaso, nationality, Van Gogh)*, given a positive triplet *(Pablo Picaso, nationality, Spain)*. Table 11 shows the statistics of the standard datasets that we used for evaluating models on the triplet classification subtask.

5.3.2. Metric

We use three metrics , i.e. *Classification Accuracy*, *Precision-recall Curve* and *Area Under Curve (AUC)*, to measure the performances among the competing methods.

DATASET	NELL-50K	WN-100K	FB-500K	NELL-1M
#(ENTITIES)	29,904	38,696	14,951	82,691
#(RELATIONS)	233	11	1,345	218
#(TRAINING EX.)	57,356	112,581	483,142	1,000,000
#(TC VALIDATING EX.)	21,420	10,436	100,000	49,728
#(TC TESTING EX.)	21,412	42,176	118,142	49,714

Table 11: Statistics of the datasets used for the triplet classification task.

- *Classification Accuracy*: We sum up the correctness of each triplet $\langle h, r, t \rangle$ via comparing the probability of the triplet and the relation-specific threshold σ_r, which can be searched via maximizing the classification accuracy on the validation triplets which belong to the relation r.

- *Precision-recall Curve*: It measures the global performance of classification by sorting all the triplets based on their estimated probability. We consider the positive testing triplets and draw the precision-recall curve for each approach.

- *Area Under Curve (AUC)*: The AUC is a commonly used evaluation metric for binary classification problems like predicting a Buy or Sell decision (binary decision). The interpretation here is that given a random positive triplet and a negative triplet, the AUC gives the proportion of the time we guess which is which correctly. It is less affected by sample balance than accuracy. A perfect model will score an AUC of 1.0, while random guessing will score an AUC of around 0.5, a meager 50% chance on each other.

5.3.3. Performance

We use the best combination of parameter settings in the entity inference task: $d = 100$, $\gamma = 0.01$, $norm = L_2$ to generate the entity and relation embeddings, and learn the best classification threshold σ_r for each relation r.

DATASET	NELL-50K	WN-100K	FB-500K	NELL-1M
METRIC	*ACC.*	*ACC.*	*ACC.*	*ACC.*
TransE [13]	80.5%	64.2%	79.9%	64.0%
TransM [30]	82.0%	57.2%	85.8%	64.8%
TransH [31]	83.6%	59.5%	87.7%	67.0%
PBE	**90.2%**	**67.8%**	**92.6%**	**86.2%**

Table 12: The accuracy of triplet classification compared among several latest approaches: *PBE*, *TransH*, *TransM* and *TransE*.

DATASET	NELL-50K	WN-100K	FB-500K	NELL-1M
METRIC	*AUC*	*AUC*	*AUC*	*AUC*
TransE [13]	0.623	0.674	0.645	0.547
TransM [30]	0.683	0.610	0.772	0.558
TransH [31]	0.681	0.613	0.744	0.596
PBE	**0.942**	**0.786**	**0.936**	**0.786**

Table 13: The AUC of triplet classification compared among several latest approaches: *PBE*, *TransH*, *TransM* and *TransE*.

Compared with several of the latest approaches, i.e. *TransH* [31], *TransM* [30] and *TransE* [13], the proposed *PBE* approach still outperforms them within the metrics of *Classification Accuracy (ACC.)* and *Area Under Curve (AUC)*, as shown in Table 12 and 13. We also draw the precision-recall curves which indicate the capability of global discrimination by ranking the distance of all the testing triplets, and Figure 2, 3, 4 and 5 can intuitively show that *PBE* performs much better than the other approaches.

Compared with several of the latest approaches, i.e. *TransH* [31], *TransM* [30] and *TransE* [13], the proposed *PBE* approach outperforms with the relative improvements that

- **NELL-50K**: {*Accuracy*: 7.9% ⇑, *AUC*: 37.9% ⇑};
- **WN-100K**: {*Accuracy*: 5.6% ⇑, *AUC*: 16.6% ⇑};
- **FB-500K**: {*Accuracy*: 5.6% ⇑, *AUC*: 21.2% ⇑};
- **NELL-1M**: {*Accuracy*: 28.6% ⇑, *AUC*: 31.8% ⇑}.

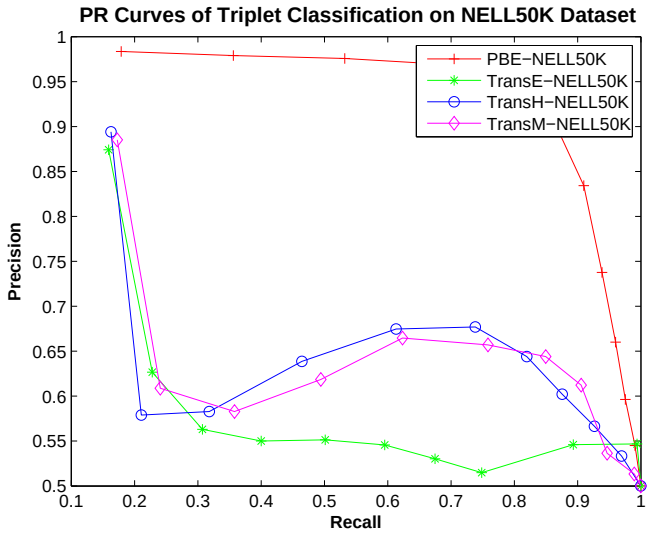

Figure 2: The precision-recall curves for triplet classification of *PBE*, *TransH*, *TransM* and *TransE* on **NELL-50K** dataset.

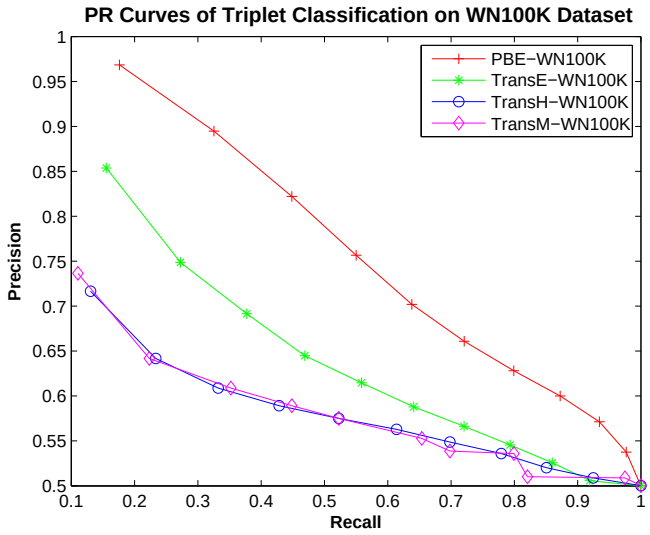

Figure 3: The precision-recall curves for triplet classification of *PBE*, *TransH*, *TransM* and *TransE* on **WN-100K** dataset.

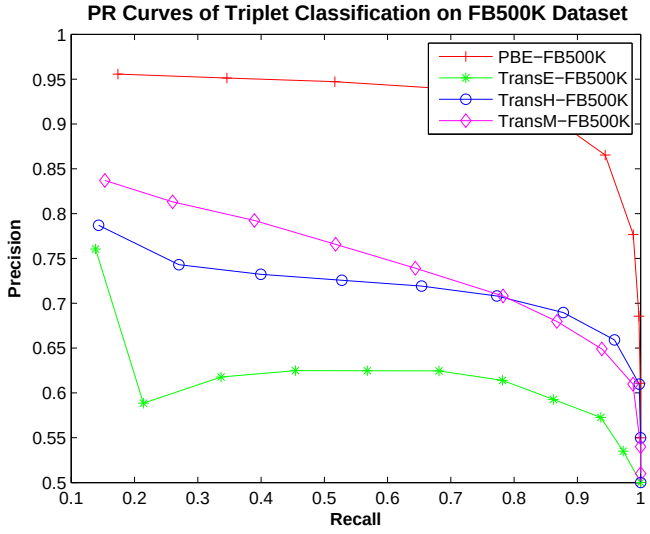

Figure 4: The precision-recall curves for triplet classification of *PBE*, *TransH*, *TransM* and *TransE* on **FB-500K** dataset.

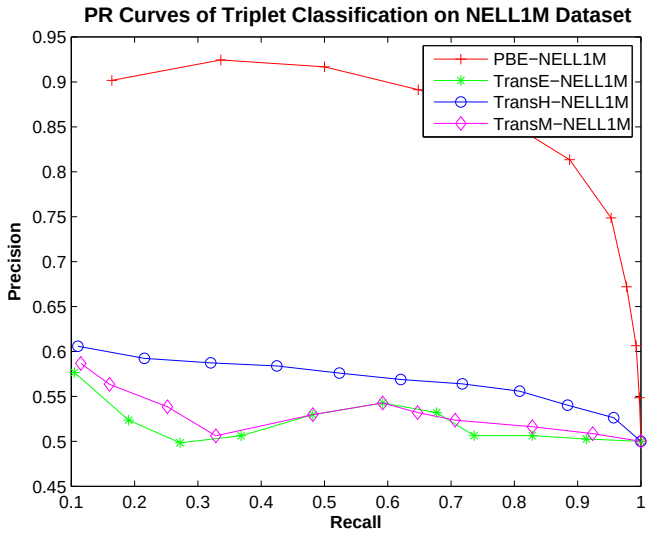

Figure 5: The precision-recall curves for triplet classification of *PBE*, *TransH*, *TransM* and *TransE* on **NELL-1M** dataset.

6. Conclusion

We challenge the problem of embedding beliefs which both contain structured knowledge and unstructured free texts and propose an elegant probabilistic model to tackle this issue at the first attempt by measuring the probability of a given belief $\langle h, r, t, m \rangle$. To efficiently learn the embeddings for each entity, relation, and word in mentions, we also adopt the negative sampling technique to transform the original model and display the algorithm based on stochastic gradient descend (SGD) to search the optimal solution. Extensive experiments on knowledge population including *entity inference*, *relation prediction* and *triplet classification* show that our approach achieves significant improvement on three large-scale repositories, compared with state-of-the-art methods. You can access to all the datasets through the publish link of OneDrive: `http://1drv.ms/1IDwZAR`.

We are pleased to see further improvements of the proposed model, which leaves open promising directions for the future work, such as taking advantage of

the probabilistic belief embeddings to enhance the studies of text summarization and open-domain question answering.

Acknowledgement

The paper is dedicated to all the members of CSLT[10] and Proteus Group [11]. It was supported by National Program on Key Basic Research Project (973 Program) under Grant 2013CB329304, National Science Foundation of China (NSFC) under Grant No.61433018 and No.61373075, and Chinese Scholarship Council, when the first author was a joint-supervision Ph.D. candidate of Tsinghua University and New York University.

[10] http://cslt.riit.tsinghua.edu.cn/

[11] http://nlp.cs.nyu.edu/index.shtml

References

[1] R. Grishman, Information extraction: Techniques and challenges, in: International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, SCIE '97, Springer-Verlag, London, UK, UK, 1997, pp. 10–27.
URL http://dl.acm.org/citation.cfm?id=645856.669801

[2] S. Sarawagi, Information extraction, Foundations and trends in databases 1 (3) (2008) 261–377.

[3] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (11) (1995) 39–41.

[4] K. Bollacker, R. Cook, P. Tufts, Freebase: A shared database of structured general human knowledge, in: AAAI, Vol. 7, 2007, pp. 1962–1963.
URL http://www.aaai.org/Papers/AAAI/2007/AAAI07-355.pdf

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008, pp. 1247–1250.
URL http://dl.acm.org/citation.cfm?id=1376746

[6] G. Klyne, J. J. Carroll, Resource description framework (rdf): Concepts and abstract syntax, W3C Recommendation.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., T. M. Mitchell, Toward an architecture for never-ending language learning, in: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010), 2010.

[8] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, D. Lin, Knowledge base completion via search-based question answering, in: WWW, 2014.
URL http://www.cs.ubc.ca/~murphyk/Papers/www14.pdf

[9] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, Association for Computational Linguistics, 2009, pp. 1003–1011.

[10] M. Fan, D. Zhao, Q. Zhou, Z. Liu, T. F. Zheng, E. Y. Chang, Distant supervision for relation extraction with matrix completion, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 839–849.
URL http://www.aclweb.org/anthology/P14-1079

[11] A. Bordes, J. Weston, R. Collobert, Y. Bengio, et al., Learning structured embeddings of knowledge bases., in: AAAI, 2011.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/viewPDFInterstitial/3659/3898

[12] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, Machine Learning 94 (2) (2014) 233–259.
URL http://link.springer.com/article/10.1007/s10994-013-5363-6

[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.

[14] J. Weston, A. Bordes, O. Yakhnenko, N. Usunier, Connecting language and knowledge bases with embedding models for relation extraction, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1366–1371.
URL http://www.aclweb.org/anthology/D13-1136

[15] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph and text jointly embedding, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1591–1601.
URL http://aclweb.org/anthology/D14-1167

[16] M. Fan, K. Cao, Y. He, R. Grishman, Jointly embedding relations and mentions for knowledge population, arXiv preprint arXiv:1504.01683.

[17] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, in: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Association for Computational Linguistics, 2004, p. 22.

[18] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, The Journal of Machine Learning Research 3 (2003) 1083–1106.

[19] Z. GuoDong, S. Jian, Z. Jie, Z. Min, Exploring various knowledge in relation extraction, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 427–434. doi:10.3115/1219840.1219893.
URL http://dx.doi.org/10.3115/1219840.1219893

[20] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 148–163.

[21] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D. S. Weld, Knowledge-based weak supervision for information extraction of overlapping relations, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 541–550.

[22] M. Surdeanu, J. Tibshirani, R. Nallapati, C. D. Manning, Multi-instance multi-label learning for relation extraction, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 455–465.

[23] J. R. Quinlan, R. M. Cameron-Jones, Foil: A midterm report, in: Proceedings of the European Conference on Machine Learning, ECML '93, Springer-Verlag, London, UK, UK, 1993, pp. 3–20.
URL http://dl.acm.org/citation.cfm?id=645323.649599

[24] N. Lao, W. W. Cohen, Relational retrieval using a combination of path-constrained random walks, Machine learning 81 (1) (2010) 53–67.

[25] N. Lao, T. Mitchell, W. W. Cohen, Random walk inference and learning in a large scale knowledge base, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 529–539.
URL http://www.aclweb.org/anthology/D11-1049

[26] M. Gardner, P. P. Talukdar, B. Kisiel, T. M. Mitchell, Improving learning and inference in a large knowledge-base using latent syntactic cues., in: EMNLP, ACL, 2013, pp. 833–838.
URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#GardnerTKM13

[27] R. Socher, D. Chen, C. D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: Advances in Neural Information Processing Systems, 2013, pp. 926–934.

[28] I. Sutskever, R. Salakhutdinov, J. B. Tenenbaum, Modelling relational data using bayesian clustered tensor factorization., in: NIPS, 2009, pp. 1821–1828.

[29] R. Jenatton, N. Le Roux, A. Bordes, G. Obozinski, et al., A latent factor model for highly multi-relational data., in: NIPS, 2012, pp. 3176–3184.

[30] M. Fan, Q. Zhou, E. Chang, T. F. Zheng, Transition-based knowledge graph embedding with relational mapping properties, in: Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation, 2014, pp. 328–337.

[31] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., 2014, pp. 1112–1119.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531

[32] M. Fan, Q. Zhou, T. F. Zheng, Learning embedding representations for knowledge inference on imperfect and incomplete repositories, arXiv preprint arXiv:1503.08155.

[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, 2013, pp. 3111–3119.

[34] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., T. M. Mitchell, Toward an architecture for never-ending language learning, in: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010), 2010.

[35] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.