Neurocomputing 294 (2018) 12–18

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Bilinear joint learning of word and entity embeddings for Entity Linking

Hui Chen[a], Baogang Wei[a,*], Yonghuai Liu[b], Yiming Li[a], Jifang Yu[a], Wenhao Zhu[c]

[a] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, PR China
[b] Department of Computer Science, Aberystwyth University, Ceredigion SY23 3DB, UK
[c] School of Computer Engineering and Science, Shanghai University, Shanghai 200000, PR China

ARTICLE INFO

Article history:
Received 4 August 2017
Revised 12 October 2017
Accepted 29 November 2017
Available online 5 December 2017

Communicated by Dr. Haijun Zhang

Keywords:
Entity Linking
Embedding model
Learning to rank
Entity disambiguation

ABSTRACT

Entity Linking (EL) is the task of resolving mentions to referential entities in a knowledge base, which facilitates applications such as information retrieval, question answering, and knowledge base population. In this paper, we propose a novel embedding method specifically designed for EL. The proposed model jointly learns word and entity embeddings which are located in different distributed spaces, and a bilinear model is introduced to simulate the interaction between words and entities. We treat EL as a ranking problem, and utilize a pairwise learning-to-rank framework with features constructed with learned embeddings as well as conventional EL features. Experimental results show the proposed model produces effective embeddings which improve the performance of our EL algorithm. Our method yields the state-of-the-art performances on two benchmark datasets CoNLL and TAC-KBP 2010.

1. Introduction

Entity Linking (EL) aims to link mentions to referential entities in a knowledge base (KB), which is a crucial technique to discover knowledge in texts and would facilitate different applications such as information retrieval, question answering, and knowledge base population. Wikipedia not only has abundant structural information, but also includes massive unstructured text information, so it is frequently chosen as the referential KB in the previous work. We also choose Wikipedia as our referential KB in this paper.

EL is a challenging problem because human language is ambiguous. For example, the mention *Spurs* in Fig. 1 has more than ten potential entities in KB, and the correct referential entity is *San Antonio Spurs* which is a basketball team. Derived from the distributional hypothesis [1], it is supposed entities are evidenced by context they occur in. For instance, as shown in Fig. 1, context word (*championships*) and entities (*NBA, Boston Celtics, Los Angeles Lakers, Chicago Bulls*) are all strong evidences for understanding the mention *Spurs*.

In recent years, it has been growing more interest in learning distributed representation of words [2] and entities [3]. Several works have been proposed to use word and entity embeddings in EL. Huang et al. [4] propose a method to learn entity embeddings, which are used to compute entity coherence. Entities are linked by maximizing the global coherence of assigned entities. But previous work [5] shows it would be helpful to take context words into consideration. Yamada et al. [6] suppose word and entity embeddings located in the same distributed space, and propose a specific joint embedding model for EL. However, words and entities have different distributed spaces. This is based on the intuition that entities usually consist of one or more words, and the scales of words and entities in text are also different. The method in [7] maps words and entities into different distributed spaces and uses a neural tensor network to model interaction between words and entities. However, the tensor network has too many parameters, which is computationally expensive and requires more training data.

In this paper, we propose a novel bilinear joint learning model (BJLM) to learn word and entity embeddings, which are supposed to locate in different distributed spaces. BJLM extends the *skip-gram* [2] framework by injecting a bilinear model to simulate the interaction between words and entities. For ranking candidate entities, we use a pairwise boosting regression tree (PBRT) model [8], whose input includes features constructed with learned embeddings and conventional EL features.

We evaluate our method on two benchmark datasets: CoNLL and TAC-KBP 2010. Experimental results show BJLM produce effective embeddings which benefit the EL algorithm. Our proposed method outperforms the state-of-the-art methods on both datasets. Our contributions include the following three aspects:

[*] Corresponding author.
E-mail addresses: chhui@zju.edu.cn (H. Chen), wbg@zju.edu.cn (B. Wei), yyl@aber.ac.uk (Y. Liu), liyiming90@zju.edu.cn (Y. Li), impyjf@zju.edu.cn (J. Yu), whzhu@i.shu.edu.cn (W. Zhu).

https://doi.org/10.1016/j.neucom.2017.11.064
0925-2312/© 2017 Published by Elsevier B.V.

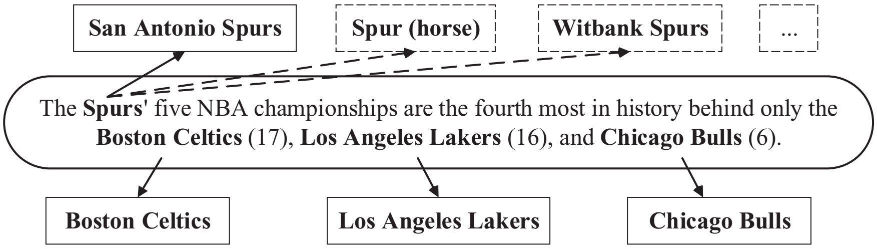
San Antonio Spurs
Spur (horse)
Witbank Spurs
...
The Spurs' five NBA championships are the fourth most in history behind only the Boston Celtics (17), Los Angeles Lakers (16), and Chicago Bulls (6).
Boston Celtics
Los Angeles Lakers
Chicago Bulls

Fig. 1. An example of Entity Linking. The bold words (**Spurs, Boston Celtics, Los Angeles Lakers, Chicago Bulls**) in the given sentence are mentions, and the corresponding potential entities are marked with boxes. Both mentions and entities usually consist of one or more words. For an instance, as for the mention **Spurs**, there list three potential entities, and the correct entity is **San Antonio Spurs** which is marked with a solid box.

(i) Our EL algorithm achieves state-of-the-art performance on two standard EL datasets. (ii) We propose a bilinear joint learning model for embeddings learning and make it open source; (iii) We investigate a boosting regression tree model with a pairwise loss function for EL.

2. Related work

Entity Linking has been widely studied in the last decade. EL algorithms are mainly divided into two categories. Algorithms in a local paradigm link entities by comparing the similarity between context information of a mention and the corresponding candidate entities in KB. Global paradigm assumes that entities occurred in the same document would have a high global coherence. Here we review some recent works related to our approach.

Traditional one-hot representation of words would come across the sparsity problem. Embedding is a popular and effective method to represent words with vectors of a specific dimension. *Skip-gram* [2] is an embedding model which aims to train word embeddings that are helpful to predict context words. Given a word w and a context word w_c, it tries to maximize to probability $P(w_c|w)$ via a softmax process. However, in order to compute $P(w_c|w)$, it needs to scan the whole vocabulary whose size is usually large, and the cost is expensive. *Skip-gram* uses negative sampling (NEG) which simplifies noise contrastive estimation (NCE) method [9] for computing the probability approximately. Our proposed embedding models are extensions of *skip-gram*.

Several graph-based approaches which are based on Wikipedia link structure are proposed. Alhelbawy and Gaizauskas [10] use the PageRank [11] algorithm to rank all entity candidates collectively. Pershina et al. [12] use the Personalized PageRank (PPR) [13] to combine local and global information. These methods construct a graph whose nodes are mention-entity pairs, and rank all nodes via a random walk process. However, it is not convenient to model various information via a single graph.

Several works use neural networks to solve EL task. Huang et al. [4] utilize a deep neural network (DNN) to learn entity embeddings, and use a semi-supervised graph regularization model to rank candidate entities collectively. Hu et al. [14] utilize Wikipedia's category as structured knowledge to improve entity embeddings, and use a model to maximize the global coherence among assigned entities. However, these methods learn entity embeddings independently without interaction with words. Yamada et al. [6] propose a joint learning method to map words and entities into the same continuous vector space, and use a gradient boosting regression trees (GBRT) model to rank candidate entities. But it is restrictive to assume words and entities are located in the same distributed space. Sun et al. [7] propose a tensor neural network to model interaction of mention, context and entity, and use a local method to rank candidate entities. However, the tensor network is computationally expensive and requires more training data. Francis-Landau et al. [15] use convolutional neural networks to model semantic correspondence between a mention's context and a candidate entity, and rank candidate entities with a final logistic regression layer.

Different from previous EL studies, we propose BJLM to learn word and entity embeddings which are located in different distributed spaces. Moreover, we investigate PBRT for ranking candidate entities and introduce some new useful features constructed on the learned embeddings.

3. Methodology of joint learning

3.1. Simplifiedjoint learning model

Words and entities are used alternately in natural language, so their embeddings should interact with each other in the learning process. It would have a semantic gap if the embeddings are learned independently [6]. We first propose a Simplified Joint Learning Model (SJLM). SJLM learns word and entity embeddings located in the same space. Formally, given a sequence of N strings consisting of words and entities $s_1, s_2, \ldots, s_N$, SJLM aims to maximize the following objective

$$\mathcal{L}_S = \sum_{i=1}^{N} \sum_{s_c \in context(s_i)} log P_S(s_c|s_i) \tag{1}$$

where s_i is the target string and $context(s_i)$ represents context strings of s_i. The conditional probability is computed as

$$P_S(s_c|s_i) = \frac{exp(e(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}))}{\sum_{s'_c \in S} exp(e(\mathbf{v}_{s_i}, \mathbf{v}_{s'_c}))} \tag{2}$$

where S is the set of all words and entities in training corpus. SJLM is similar to *skip-gram* except that it learns both word and entity embeddings.

3.2. Bilinearjoint learning model

SJLM learns word and entity embeddings in the same space. However, entities usually consist of one or more words, and the scales of words and entities in text are different. Based on this intuition, we suppose word and entity located in different distributed spaces. Inspired by the work [16], we introduce a bilinear model to simulate the interaction between two embeddings. Formally, let $\mathbf{v}_e \in R^{d_1}$ denote an entity embedding, $\mathbf{v}_w \in R^{d_2}$ denote a word embedding and $M_B \in R^{d_1 \times d_2}$ denote a projection matrix. The bilinear model is defined as $f_b(\mathbf{v}_e, \mathbf{v}_w) = \mathbf{v}_e^{\top} M_B \mathbf{v}_w$. It is noted the projection matrix M_B only works between an entity embedding and a word embedding. When the parameters of f_b are both entity or word embeddings, M_B decays to an identity matrix with the same dimension of parameters. We define the complete bilinear model

$$f(\mathbf{v}_1, \mathbf{v}_2) = \begin{cases} \mathbf{v}_1^{\top} M_B \mathbf{v}_2 & \mathcal{I}(\mathbf{v}_1, \mathbf{v}_2) = 0 \\ \mathbf{v}_1^{\top} \mathbf{v}_2 & \mathcal{I}(\mathbf{v}_1, \mathbf{v}_2) = 1 \end{cases} \tag{3}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are embeddings (entity embeddings or word embeddings), and $\mathcal{I}(\mathbf{v}_1, \mathbf{v}_2)$ is an indicator function whose value is 1

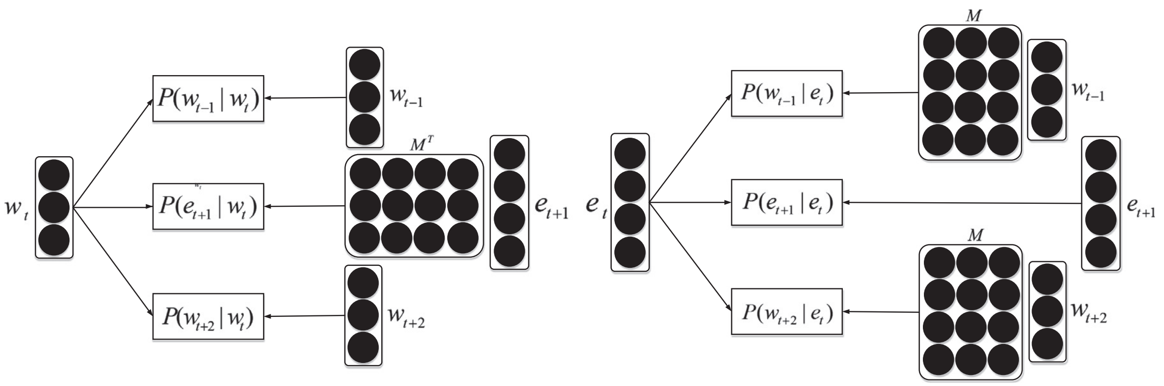

Fig. 2. Two instances of BJLM. The left one uses a target word w_t to predict context strings which contain two words w_{t-1}, w_{t+2} and an entity e_{t+1}. The right one uses a target entity e_t to predict context strings which contain two words w_{t-1} w_{t+2} and an entity e_{t+1}. When the target string and the context string are in different types of embeddings, the projection matrix is used to solve the space gap.

only when $\mathbf{v}_1$ and $\mathbf{v}_2$ are in the same distributed space. We use f to replace the energy function e in Eq. (2) for the probability computation and get

$$P_B(s_c|s_i) = \frac{exp(f(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}))}{\sum_{s'_c \in S} exp(f(\mathbf{v}_{s_i}, \mathbf{v}_{s'_c}))} \tag{4}$$

Afterwards, replacing $P_S(s_c|s_i)$ in Eq. (1) with $P_B(s_c|s_i)$, we get an objective function $\mathcal{L}_B$. Formally, given a sequence of strings $s_1, s_2, \ldots, s_N$, the objective $\mathcal{L}_B$ is defined as

$$\mathcal{L}_B = \sum_{i=1}^{N} \sum_{s_c \in context(s_i)} logP_B(s_c|s_i) \tag{5}$$

where s_i is the target string, and $context(s_i)$ represents context strings of s_i. Word and entity embeddings would be produced when maximizing the objective $\mathcal{L}_B$, and the interaction between embeddings is modeled by the projection matrix M_B. Fig. 2 shows two instances to describe how BJLM works.

3.3. Training

Because of the large number of words and entities in training corpus, the computation cost of probabilities $P_S(s_c|s_i)$ and $P_B(s_c|s_i)$ is expensive. Following [17], we also use NEG method to compute the probabilities approximately. The NEG objective of $\mathcal{L}_B$ is

$$log\sigma(f(\mathbf{v}_{s_i}, \mathbf{v}_{s_c})) \quad + \\ \sum_{i=1}^{k} E_{s'_c} \sim P_n(s_i)[log\sigma(-f(\mathbf{v}_{s_i}, \mathbf{v}_{s'_c}))] \tag{6}$$

Since the parameters of function f include word and entity embeddings, the corresponding derivatives are

$$\frac{\partial f}{\partial \mathbf{v}_{s_i}} = \begin{cases} \mathbf{v_{s_c}} & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 1 \\ M_B \mathbf{v}_{s_c} & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 0 \end{cases} \tag{7}$$

$$\frac{\partial f}{\partial \mathbf{v}_{s_c}} = \begin{cases} \mathbf{v_{s_i}} & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 1 \\ M_B^\top \mathbf{v}_{s_i} & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 0 \end{cases} \tag{8}$$

$$\frac{\partial f}{\partial M_B} = \begin{cases} 0 & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 1 \\ \mathbf{v}_{s_i}^\top \mathbf{v}_{s_c} & \mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}) = 0 \end{cases} \tag{9}$$

where $\mathcal{I}(\mathbf{v}_{s_i}, \mathbf{v}_{s_c})$ is the indicator function mentioned in Section 3.2. We use stochastic gradient descent (SGD) to train Eq. (6), and the parameter updating formulas are

$$\begin{aligned} \mathbf{v}_{s_i} &= \mathbf{v}_{s_i} + \eta * [l - \sigma(f(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}))] \frac{\partial f}{\partial \mathbf{v}_{s_i}} \\ \mathbf{v}_{s_c} &= \mathbf{v}_{s_c} + \eta * [l - \sigma(f(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}))] \frac{\partial f}{\partial \mathbf{v}_{s_c}} \\ M_B &= M_B + \eta * [l - \sigma(f(\mathbf{v}_{s_i}, \mathbf{v}_{s_c}))] \frac{\partial f}{\partial M_B} \end{aligned} \tag{10}$$

where $\mathbf{v}_{s_i}$ is an embedding of the target string, $\mathbf{v}_{s_c}$ is an embedding of the context string, η is the learning rate and l denotes a label whose value is 1 when s_c is a true context string, otherwise 0 when s'_c is sampled from the noise distribution $P_n(s_i)$. This noise distributions of words and entities are both unigram distributions raised to the 3/4th power [17].

4. Entity Linking

Given a document d containing a set of pre-tagged mentions $\{m_1, m_2, \ldots, m_N\}$, EL aims to find a set of referential entities $\{e_1, e_2, \ldots., e_N\}$ in KB. Generally, EL could be divided into two steps: candidate entity generation and ranking. Candidate entities are usually generated based on a dictionary. We mainly describe the ranking algorithm in this section. Algorithm 1 shows a skeleton of our EL method.

Algorithm 1: The skeleton of our EL method. In step 2, c_i represents a list of candidate entities of corresponding mention m_i.

```
  Input: mention set $M = \{m_1, m_2, \ldots, m_n\}$
1 Learning word and entity embeddings
2 Generating candidate entity set $C = \{c_1, c_2, \ldots., c_n\}$
3 Constructing features of each pair $(m_i, c_i^k)$, $m_i \in M$, $c_i^k \in c_i$,
  $c_i \in C$
4 Training PBRT with constructed features on training dataset
5 Applying trained PBRT on test dataset
  Output: predicted entity set $E$
```

4.1. Pairwise ranking model

EL could be treated as a learning to rank (L2R) problem. Given a query mention, a ranking algorithm assigns each candidate entity a ranking score, and the one with the highest score is chosen as the referential entity. However, we do not intend to investigate various L2R methods for EL in this paper. Shen et al. [18] treat EL as a pairwise ranking problem, and put forward a method based on ranking SVM. Yamada et al. [6] use GBRT with a pointwise loss function to rank candidate entities. Given a mention, it would usually generate many candidate entities but at most one true referential entity, so pointwise ranking methods may come across the label bias problem. In this paper, we adopt a supervised model PBRT for ranking candidate entities, and construct a set of features which would be introduced in the next section.

4.2. Features

4.2.1. Entityprior probability

Each mention m_i has a set of candidate entities $\{c_i^1, c_i^2, \ldots, c_i^k\}$. Each candidate entity c_i^k is assigned with a prior probability which is computed as $P(c_i^k|m_i) = |m_i \rightarrow c_i^k|/|m_i|$, where $|m_i|$ represents the frequency of anchors with the same surface form as mention m_i and $|m_i \rightarrow c_i^k|$ is the count of m_i linked to candidate entity c_i^k.

4.2.2. Textual context

Given a mention m_i, its textual context vector $\mathbf{v}_{tc}(m_i)$ is computed as the average of context word embeddings

$$\mathbf{v}_{tc(m_i)} = \frac{1}{|tc(m_i)|} \sum_{w \in tc(m_i)} \mathbf{v}_w \tag{11}$$

where $tc(m_i)$ represents textual context of m_i and we use all noun words in current document. As SJLM and BJLM both produce embeddings, there are two types of similarities. Given a candidate entity c_i^k of m_i, textual context similarity between c_i^k and m_i is computed as

$$sim(m_i, c_i^k) = \begin{cases} \mathbf{v}_{\mathbf{c_i^k}}^\top \mathbf{v}_{\mathbf{tc(m_i)}} & for \quad SJLM \\ \mathbf{v}_{c_i^k}^\top M_B \mathbf{v}_{tc(m_i)} & for \quad BJLM \end{cases} \tag{12}$$

4.2.3. Entity coherence

Entity coherence has been shown as an effective feature in previous work [19], but exhaustive entity coherence computation is a NP hard problem. We pick out unambiguous entities whose embeddings are averaged to derive the context entity vector

$$\mathbf{v}_{ec(m_i)} = \frac{1}{|ec(m_i)|} \sum_{e \in ec(m_i)} \mathbf{v}_e \tag{13}$$

where $ec(m_i)$ represents context entities of m_i and we use unambiguous entities. We consider an entity unambiguous if its prior probability is greater than 0.95. Given a candidate entity c_i^k of m_i, coherence between c_i^k and $ec(m_i)$ is computed as $coh(c_i^k, ec(m_i)) = \mathbf{v}_{c_i^k}^\top \mathbf{v}_{ec(m_i)}$.

4.2.4. Entity importance

We construct a graph G whose nodes are mention-entity pairs. Given two nodes (m_i, c_i^k) and (m_j, c_j^l), the edge weight is computed as $\mathbf{v}_{c_i^k}^\top \mathbf{v}_{c_j^l}$. Two nodes are connected in G only when the candidate entities are linked in Wikipedia link structure. We run the PPR algorithm [13] on G. Each node is assigned with a score which is treated as an importance score for each candidate entity embedded in current node.

4.2.5. Saliententity support

Human beings usually pay more attention to some key context words or entities when understanding an entity in a document, and attention is one such mechanism. Globerson et al. [20] introduce a coherence model with a multi-focal attention mechanism. We design a feature to model information of the K most relative context entities via an Integer Linear Programming (ILP) model, which could be considered as a hard attention constraint on context entities.

$$\sum_i \sum_k [\alpha x_i^k sim(m_i, c_i^k) + \beta \sum_{j \neq i} y_{i,k}^j ss_{m_j}(c_i^k)] \tag{14}$$

$$\begin{aligned} s.t. \quad & x_i^k \in \{0,1\} && , \quad \forall_i \sum_k x_i^k = 1 \\ & y_{i,k}^j \in \{0,1\} && , \quad \forall_{i,k} \sum_j y_{i,k}^j = K \\ & y_{i,k}^j \leq x_i^k && , \quad \alpha + \beta = 1 \end{aligned} \tag{15}$$

where x_i^k and $y_{i,k}^j$ are binary variables to be solved. Score $sim(m_i, c_i^k)$ is textual context similarity. Score $ss_{m_j}(c_i^k)$ is computed as $(\sum_l pp(c_j^l)\mathbf{v}_{c_j^l}^\top \mathbf{v}_{c_i^k})/|c_j^l|$, where $pp(c_j^l)$ is the prior probability to choose candidate entity c_j^l given mention m_j, and $|c_j^l|$ denotes the total number of candidate entities of m_j. In experiments, K is tuned to 3, α is set to 0.3 and β is set to 0.7 empirically.

4.2.6. Other features

In addition to above-mentioned features, we also use some features referring to [6], including Wikipedia entity popularity, the maximum prior probability of a candidate entity of all mentions in current document, the number of candidate entities for a mention. Let S_{m_i} denote the surface of mention m_i, and $T_{c_i^k}$ denote the title of candidate entity c_i^k. We construct several string features including edit distance between S_{m_i} and $T_{c_i^k}$, whether $T_{c_i^k}$ equals to or contains S_{m_i}, whether $T_{c_i^k}$ starts with or ends with S_{m_i}.

5. Experiments

In this section, we describe experimental settings and results. We first illustrate how to train the proposed embedding models. Then, we describe experimental details on two standard EL datasets. Finally, we give an analysis of experimental results.

5.1. Training for embedding models

SJLM and BJLM are both trained on English Wikipedia dump (20151102 version). We use JWPL [21] toolkit to parse dump files. Redirect and disambiguation pages are removed. All page titles are treated as referential entities in KB. All anchors in pages are replaced with titles via Wikipedia links, and all numbers are spelt out. After a pre-processing step, we obtain about 2 billion words and 63 million entities. Words occurred less than five times are discarded during training process. Finally we learn about 2.4 million word embeddings and 3 million entity embeddings.

Most parameters of SJLM and BJLM are the same. The context window size is 10, and the number of negative samples is 15. Both models iterate once in the given training corpus with a learning rate 0.025. The noise distributions of words and entities are both unigram distributions raised to the 3/4th power. The word embedding dimension is set to 100. The only different parameter is entity embedding dimension which is set to 120 for BJLM while 100 for SJLM. In addition, BJLM has a projection matrix $M_B \in R^{120 \times 100}$. It costs almost 12 h for training BJLM and 10 h for training SJLM without GPU Acceleration. Here is our training machine : Intel(R) Xeon(R) CPU E5-2620.

5.2. Entity Linking

5.2.1. Evaluation dataset

CoNLL: The CoNLL dataset [5] consists of training, development and test sets, which contains 946, 215 and 231 documents respectively. Each occurrence of a mention is annotated with an entity or

Table 1
Statistics information of candidate entity generation on CoNLL dataset. DS is the size of the dataset. M is the total number of mentions. MR is the count of mentions with at least an candidate entity. GR is the count of mentions with a valid candidate entity. U is the count of mentions with only one candidate entity. CCE is the count of all candidate entities for all mentions.

	DS	M	MR	GR	U	CCE
Training	946	23,396	18,425	17,677	4322	238,855
Development	215	5904	4773	4509	1115	63,276
Test	231	5616	4451	4321	961	63,345

Table 2
Statistics information of candidate entity generation on TAC-KBP 2010 dataset.

	DS	M	MR	GR	U	CCE
Training	1453	1500	1074	1028	147	24242
Test	2231	2250	1018	953	130	20429

a NIL. We report the standard micro- and macro- accuracies of the top-ranked candidate entities on the test dataset. We use a publicly available dictionary [12] to generate candidate entities.

TAC-KBP 2010: The TAC-KBP 2010 dataset [22] consists of 1453 training document and 2231 test documents. Most documents contain one query mention annotated with an entity or a NIL. We report the micro-accuracies of the top-ranked candidate entities on the test dataset. We utilize the Stanford NER [23] toolkit to recognize named entities in documents as context query mentions. A key-value dictionary is constructed from Wikipedia's articles, redirect pages and disambiguation pages for candidate entity generation. All anchors are extracted as keys, and corresponding Wikipedia titles are extracted as values.

5.2.2. KB and candidate entity generation

We use the 20151102 version of Wikipedia as our KB. Mentions in TAC-KBP 2010 dataset are annotated to Freebase, and we map the annotations to our KB. Due to the evolution of Wikipedia, some annotated entities are already not available in KB on both datasets. We retain the mentions with valid entities for evaluation. A third dictionary is used to generate candidate entities on CoNLL dataset, and Table 1 shows the statistics information. For TAC-KBP 2010, we generate candidate entities with a constructed dictionary, and retain the top 50 candidate entities ranked by prior probability. Table 2 shows the statistics information of TAC-KBP 2010 dataset.

As shown in two tables, about 20.7% of mentions on CoNLL test dataset and 54.8% of mentions on TAC-KBP 2010 test dataset are not linkable in the KB. The third dictionary gets a recall of 0.971 on the CoNLL test dataset, and the averaged ambiguity (CCE/GR) is 14.66. Our constructed dictionary gets a recall of 0.936 on TAC-KBP 2010 test dataset with a averaged ambiguity 21.4.

5.2.3. Baseline models

We put forward a baseline EL model PBRT$_S$, which is trained with features constructed with embeddings learned by SJLM. Besides PBRT$_S$, we choose other three state-of-the-art methods as baselines.

- Gloverson et al. [20] put forward a coherence model with a multi-focal attention mechanism.
- PPRSim [12] is a graph-based EL approach based on Personalized PageRank.
- Yamada et al. [6] propose a joint embedding model, and utilize a GBRT model to rank candidate entities.

5.2.4. Experimental results

We propose an EL model PBRT$_B$. Different from PBRT$_S$, PBRT$_B$ is trained on features constructed with embeddings learned by BJLM. Table 3 shows experimental results on two datasets. PBRT$_B$ achieves a micro-accuracy of 0.938 and macro-accuracy of 0.935 on CoNLL dataset, and micro-accuracy of 0.881 on TAC-KBP 2010 dataset. It shows PBRT$_B$ outperforms baselines on both datasets.

Table 3
Results on CoNLL and TAC-KBP 2010 test datasets.

	CoNLL (micro)	CoNLL (macro)	TAC10 (micro)
PBRT$_B$	**0.938**	**0.935**	**0.881**
PBRT$_S$	0.932	0.929	0.865
Yamada et al. [6]	0.931	0.926	0.855
PPRSim	0.918	0.899	–
Gloverson et al. [20]	0.927	–	0.872

5.3. Analysis

In this section, we analyse experimental results, prediction errors, features and parameters. We present a detailed analysis on CoNLL dataset. As for TAC-KBP 2010 dataset, we draw similar conclusion and would not go into details.

5.3.1. Result analysis

As shown in Table 3, PBRT$_B$ outperforms PBRT$_S$ with small margins, however the results are statistically significant. For further comparison of the embeddings learned by BJLM and SJLM, we conduct two experiments on CoNLL dataset. We train two PBRT models with only two features: *textual context* and *salient entity*, which are both directly constructed with embeddings. We achieve micro-accuracies of 0.900 for the BJLM based model and 0.891 for the SJLM based model. BJLM maps entities and words into different distribution spaces, so there are three types of information interaction during the embedding training process: word-word, entity-entity and word-entity. As for SJLM, entity and word embeddings interact with each other directly. Compared with SJLM, BJLM learns more fine-grained and representative embeddings which improve EL performance.

In order to explore the function of the pairwise loss objective, we run a pointwise boosting regression tress model with the same input features as PBRT$_B$. Finally we achieve a micro-accuracy of 0.927 which is one percentage worse than the result of PBRT$_B$. The result shows a pairwise objective works better than a pointwise objective, because the latter may encounter the label bias problem. We also modify Yamada et al.'s method with a pairwise loss objective, and achieve a micro-accuracy of 0.930, which performs a little worse than PBRT$_S$. SJLM and Yamada et al.'s embedding method have a similar learning mechanism, and the difference of performance is mainly caused by the different used features. Gloverson et al.'s method and PPRSim both only utilize entity coherence without considering textual information, which perform relatively poor.

5.3.2. Error analysis

We divide errors into three categories, and pick out typical errors for analysis.

Errors of first type are caused by coarse-grained context words. For example, given a sentence with a mention *Japan*: "*Late goals give Japan win over Syria.*", the true entity is *Japan national football team*, while the prediction is *Japan*. Human beings could understand the entity via context words, e.g., *goals* and *win*. We computed similarities between two words and candidate entities: $sim(Japan, goals) = 5.01$ and $sim(Japan, win) = 4.49$. For entity *Japan national football team*, similarities are: 15.08 and 12.31 respectively. As expected, the latter entity gets higher scores. However, the *textual context* feature uses all noun words, which is more

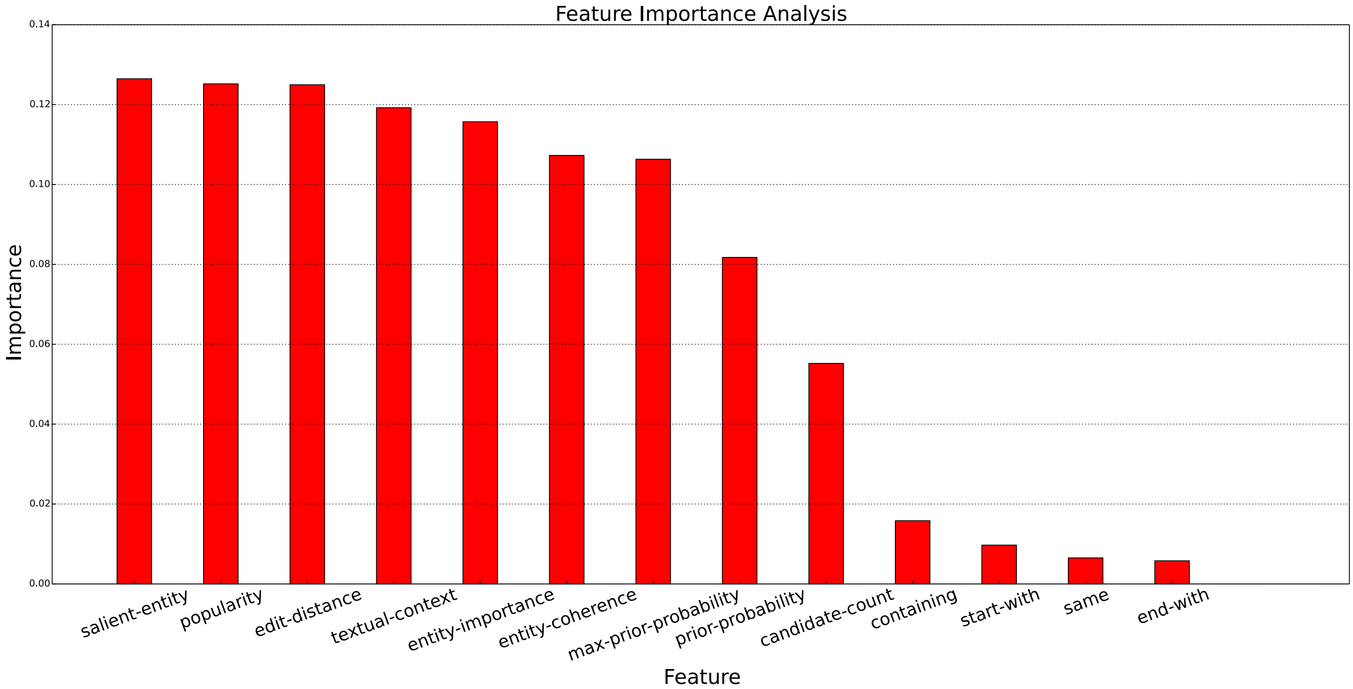

Fig. 3. Feature importance analysis of PBRT_B on CoNLL/AIDA dataset. Features are listed on the horizontal axis in a descending order ranked by their corresponding importance scores.

coarse-grained than local context words and may introduce noise. What's more, the prior probability of *Japan* is high enough (0.98) to affect the prediction.

Errors of the second type are caused by common sense. For instance, a sentence *"Barbarians-15- Tim Stimpson(England) ..."*, the mention *England* has two candidate entities *England national rugby union team* and *England*. Context information is limited here, but we could infer the sentence talks about nationality when *England* comes after a name. It is a challenge for computers to understand common sense.

Errors of the last type are also caused by common sense, but more difficult. Considering the mention *Mexico* in the sentence *"Santa Fe has mining and exploration operations in Nevada, California, Montana, Canada, Brazil, Australia, Chile, Kazakstan, Mexico and Ghana."*, it is even hard for human beings to tell whether the mention represents *Mexico* or *Mexico City* without external information, and both candidate entities are rational. Considering the preceding *Canada*, it is more likely to choose *Mexico* rather than *Mexico City*. However, this process of using common sense for decision is a challenge for computers.

We consider it feasible to introduce an attention mechanism to model context information to solve the first type of errors, and we would study this in our future work. However, it is more challenging to solve the last two types of errors, which need resort to common sense to understand sentences more appropriately.

5.3.3. Feature and parameter analysis

We analyse the importance of different features used in PBRT_B. We first compute the number of correct predictions for every single feature, and then normalize these numbers to 1. Fig. 3 shows feature importance scores.

The most important feature is *salient entity*. It is rational for human-beings to understand an entity with only considering a few relative entities while not all other entities in a document. Feature *salient entity* is mainly designed with this intuition, and it is similar to the attention mechanism. As shown in Fig. 3, some conventional features (popularity, edit-distance, max-prior-probability) are useful, but the traditional dominant feature *prior probability* is weakened in our method. The least important features are string-boolean features, and *edit distance* feature simulates the function of string-boolean features to some degree. Words are of equal importance and entities are weighted by prior probabilities in current constructed features. However, this kind of weighting mechanism is coarse-grained. We consider it more appropriate to learn weights of context words and entities via a learning process.

We investigate two parameters: embedding dimension and salient entity number *K*. We run PBRT_B based on three groups of embeddings: (a) word embedding / 50 and entity embedding / 60; (b) word embedding / 100 and entity embedding / 120; (c) word embedding / 200 and entity embedding / 250, and find that there are no significant difference of the experimental results, which illustrates embedding dimension is not a key factor in the final performance.

We conduct experiments with salient entity number $K = (1, 3, 5, 10)$, and find $K = 3$ works best. It is intuitively rational because it is enough to refer to three context entities for human beings to understand an entity. More context entities are redundant, and sometimes would introduce noise instead.

6. Conclusion

In this paper, we propose a novel bilinear joint learning model (BJLM). BJLM simultaneously learns word and entity embeddings which are located in different distributed spaces, and uses a bilinear mapping to solve the semantic gap. The learned embeddings are used to construct a series of features which are fed to PBRT together with conventional EL features. Each candidate entity is assigned with a ranking score, and the entity with the highest score is chosen as the true referential entity. We achieve the state-of-the-art results on two standard EL datasets. Experimental results show BJLM produces effective embeddings which improve the performance of our EL method.

However, we also come across some challenges. First, it is more appropriate to weight context words and entities through a learning process instead of empirical ways. Second, in our experiments the dataset scale is small, and the documents are of normal length.

It would come across new problems if datasets have large scale or the documents are very lengthy (e.g., e-books) [24]. We would study these challenges in our future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China [No. 61572434], the China Knowledge Centre for Engineering Sciences and Technology [No. CKCEST-2017-1-2], and Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015).

References

[1] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.
[2] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arxiv: 1301.3781 (2013).
[3] Y. Wu, T. Mu, J.Y. Goulermas, Translating on pairwise entity space for knowledge graph embedding, Neurocomputing 260 (2017) 411–419.
[4] H. Huang, L. Heck, H. Ji, Leveraging deep neural networks and knowledge graphs for entity disambiguation, (2015) arxiv:1504.07678.
[5] J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum, Robust disambiguation of named entities in text, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 782–792.
[6] I. Yamada, H. Shindo, H. Takeda, Y. Takefuji, Joint learning of the embedding of words and entities for named entity disambiguation, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 250–259.
[7] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, X. Wang, Modeling mention, context and entity with neural networks for entity disambiguation, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 2015, pp. 1333–1339.
[8] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the Twenty-Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[9] M.U. Gutmann, A. Hyvärinen, Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics, J. Mach. Learn. Res. 13 (1) (2012) 307–361.
[10] A. Alhelbawy, R.J. Gaizauskas, Graph ranking for collective named entity disambiguation, in: Proceedings of the Fifty-Second Annual Meeting of the Association for Computational Linguistics, 2014, pp. 75–80.
[11] L. Page, S. Brin, R. Motwani, T. Winograd, in: The Pagerank Citation Ranking: Bringing Order to the Web, Stanford InfoLab, 1999.
[12] M. Pershina, Y. He, R. Grishman, Personalized page rank for named entity disambiguation, in: Proceedings of the Annual Conference of the North American Chapter of the ACL, NAACL HLT, 14, 2015, pp. 238–243.
[13] G. Jeh, J. Widom, Scaling personalized web search, in: Proceedings of the Twelfth International Conference on World Wide Web, 2003, pp. 271–279.
[14] Z. Hu, P. Huang, Y. Deng, Y. Gao, E.P. Xing, Entity hierarchy embedding, in: Proceedings of the Fifty-Third Annual Meeting of the Association for Computational Linguistics and the Seventh International Joint Conference on Natural Language Processing (ACL-IJCNLP), 1, 2015, pp. 1292–1300.
[15] M. Francis-Landau, G. Durrett, D. Klein, Capturing semantic similarity for entity linking with convolutional neural networks, in: Proceedings of NAACL-HLT, 2016, pp. 1256–1261.
[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion., in: Association for the Advancement of Artificial Intelligence, (AAAI), 2015, pp. 2181–2187.
[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
[18] W. Shen, J. Wang, P. Luo, M. Wang, Linden: Linking Named Entities with Knowledge Base via Semantic Knowledge, in: Proceedings of the 21st international conference on World Wide Web, ACM, 2012, pp. 449–458.
[19] D. Milne, I.H. Witten, Learning to link with Wikipedia, in: Proceedings of the Seventeenth ACM Conference on Information and Knowledge Management, 2008, pp. 509–518.
[20] A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringgaard, F. Pereira, Collective entity resolution with multi-focal attention, in: Proceedings of the Fifty-Fourth Annual Meeting of the Association for Computational Linguistics, 2016, pp. 621–631.
[21] O. Ferschke, T. Zesch, I. Gurevych, Wikipedia revision toolkit: efficiently accessing Wikipedia's edit history, in: Proceedings of the Forty-Ninth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations, 2011, pp. 97–102.
[22] H. Ji, R. Grishman, H.T. Dang, K. Griffitt, J. Ellis, Overview of the TAC 2010 knowledge base population track, in: Proceedings of the Third Text Analysis Conference (TAC 2010), 3 (2), 2010, pp. 3–13.
[23] J.R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in: Proceedings of the Forty-Third Annual Meeting on Association for Computational Linguistics, 2005, pp. 363–370.
[24] H. Zhang, T.W. Chow, Q.J. Wu, Organizing books and authors by multilayer SOM, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2016) 2537–2550.

Hui Chen received his B.E. degree from the College of Computer Science, Xidian University, Xi'an, China in 2013. He is currently a Ph.D. candidate in the College of Computer Science and Technology, Zhejiang University. His current research interests include natural language processing, information retrieval and deep learning.

Baogang Wei received his Ph.D. degree from Northwestern Polytechnical Universtiy, China in 1997. He is currently a professor at Zhejiang University, China. His main research interests include artificial intelligence, pattern recognition, image processing, machine learning, digital library and information & knowledge management. Since 1999, he has been a member of the Chinese Association for Artificial Intelligence. So far, he has published more than 50 papers in international journals including IEEE TKDE and IEEE TVCG, and conference proceedings including AAAI, CIKM and PAKDD.

Yonghuai Liu received his Ph.D. degree from the University of Hull, United Kingdom in 2000. He is currently a professor at Aberystwyth University. His main research interests include 3D imaging, range image registration and projective registration. He is the area editor of journal Pattern Recognition Letters: an international journal. He is the associate editors of journal Neurocomputing. He is editorial board member of journal American Journal of Educational Research. He was associate editors of conferences including IEEE ICRA(17), ICRA(15) and ICRA(14). He has published more than 200 papers in international journal including IEEE TVCG, IEEE Transactions on Multimedia, and IEEE PAMI, and conference IEEE CVPR.

Yiming Li received the B.E. degree from the School of Information Science and Technology, Southwest Jiaotong University, China in 2013. He is currently a Ph.D. candidate in the College of Computer Science and Technology, Zhejiang University. His current research interests include natural language processing, representation learning and deep learning.

Jifang Yu received her B.E. degree in Computer Science and Technology at Shandong University, China in 2015. She is currently a Ph.D. candidate in the College of Computer Science and Technology, Zhejiang University. Her current research interests include natural language processing, sentiment analysis and deep learning.

Wenhao Zhu received his Ph.D. degree in Computer Science at Zhejiang University, China in 2009. During 2012 and 2013, he visited Computer Laboratory, University of Cambridge as a visiting scholar for one year. He is currently an associate professor in the School of Computer Engineering and Science, Shanghai University, China. His research is in the areas of information extraction, web data mining and recommendations.