Attention-Based Bilinear Joint Learning Framework for Entity Linking

Min Cao[1], Penglong Wang[1], Honghao Gao[2(✉)], Jiangang Shi[3], Yuan Tao[2], and Weilin Zhang[1]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
mcao@staff.shu.edu.cn, penglongwang@shu.edu.cn, zeroized@i.shu.edu.cn
[2] Computing Center, Shanghai University, Shanghai, China
gaohonghao@shu.edu.cn
[3] Shanghai Shang Da Hai Run Information System Co., Ltd, Shanghai 200444, China
lukepro@163.com

Abstract. Entity Linking (EL) is a task that links entity mentions in the text to corresponding entities in a knowledge base. The key to building a high-quality EL system involves accurate representations of word and entity. In this paper, we propose an attention-based bilinear joint learning framework for entity linking. First, a novel encoding method is employed for coding EL. This method jointly learns words and entities using an attention mechanism. Next, for ranking features, a weighted summation model is introduced to model the textual context and coherence. Then, we employ a pairwise boosting regression tree (PBRT) to rank candidate entities. As input, PBRT takes both features constructed with a weighted summation model and conventional EL features. Finally, through the experiment, we demonstrate that the proposed model learns embedding efficiently and improves the EL performance compared with other state-of-the-art methods. Our approach achieves superior result on two standard EL datasets: CoNLL and TAC 2010.

Keywords: Entity linking · Embedding model · Modeling context · Modeling coherence · Entity disambiguation

1 Introduction

Entity linking (EL) is a key technique for discovering knowledge in a text which is highly important for building Semantic Web. EL is a task to link entity mentions in text with corresponding entities in a knowledge base [1]. EL can help computers find important semantic information in sentences and determine how the meanings of words differ in different contexts, which is indispensable for helping computers understand natural language. EL has been widely adopted in applications such as information extraction, information retrieval, question answering system, and knowledge base population (KBP).

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 247–259, 2019.
https://doi.org/10.1007/978-3-030-30146-0_17

The challenge of EL is that human natural language is ambiguous. For example, in Fig. 1, more than five entities are likely to be related to mention "Bill Russell", however, the fact is that only one actual reference exists. The meaning of entity is decided by its context dynamically. For instance, as shown in Fig. 1, the context (rookie center) and mentions (Boston Celtics, Bob Cousy, Red Auerbach, NBA) are all valid basis for disambiguating the "Bill Russell" mention.

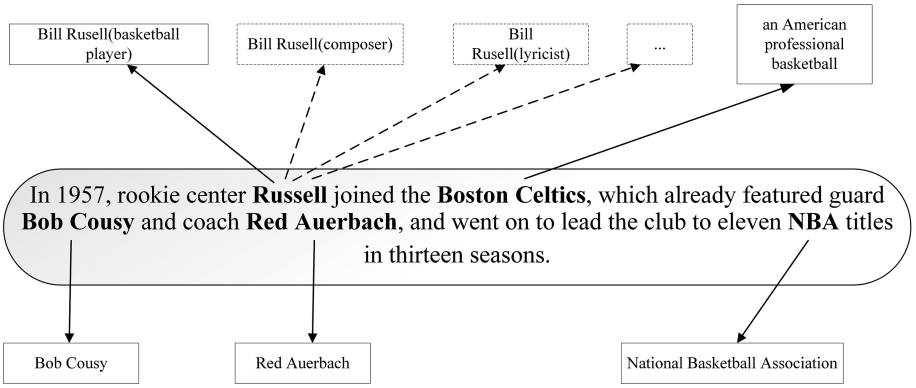

Fig. 1. An example of EL.

In recent years, the study of distributed representation for word and entity has become increasingly interested among researchers. Some works have proposed using embedding of word and entity in entity linking. Huang [2] studied entity embedding to calculate the correlation between entities. Hoffart [3] proposed taking contextual information into account. Yamada [4] assumed that word and entity are distributed in the same space and proposed a special joint learning word and entity embedding model. Chen [5] believed that words and entities should be embedded into different spaces. Hence, he developed a bilinear joint learning model (BJLM). Sun [6] put words and entities into different spaces and employed a neural tensor network to learn the interactions between word and entity. Nevertheless, none of the methods above capture different information aspects of word and entity context, which can result in a loss of information. Therefore, this paper mainly investigates how to effectively embed and combine word and entity with their context, and generates the precise embedding for these words and entities.

The semantic of a word is derived primarily from its context and relationships with other words in the same document. Most previous methods have assumed that all words and mentions in a context have the same weight. Obviously, these approaches result in bias regarding the meanings of words and mentions. In this paper, an attention-based bilinear joint learning model (ABJL) is proposed. When mapping words and mentions to different distributed spaces, ABJL focuses on the different impact of the words and mentions in the context of the target word and mention. Moreover, two EL features are constructed with learned embedding: textual context feature and entity coherence

feature. Finally, the constructed EL features as well as the traditional EL features are fed into a pairwise boosting regression tree (PBRT) [7] for candidate ranking.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 presents the bilinear joint learning method with an attention mechanism. Section 4 introduced the application of the proposed embedding method on EL task. Section 5 describes the experimental settings and result and Sect. 6 presents conclusions and provides the directions of future study.

2 Related Work

In past decades, EL has been widely studied and applied in academia. EL involves linking entity mentions in the text to corresponding entities in a knowledge base. There are three main categories of EL algorithms. First, the EL algorithms that adopt an independent paradigm use a single mention and its context information and compare its similarities with candidate entities in a knowledge base. Second, the collective EL algorithms utilize correlations between mentions in the same document to link multiple mentions to knowledge base simultaneously. Third, the collaborative EL algorithms extend contextual information associated with entity mention by means of cross-documentation and then use extended entity mention information to address EL. Here we review some recent works related to our approach.

The conventional representation approach of word named one-hot encoding encounters sparsity problems. Word-to-vector (word2vec) is an effective word representation method that has become increasingly welcome in academia. Word2vec uses a continuous vector of low-dimension to represent a word. Skip-gram [8] is another word embedding method whose goal is to train a word embedding to effectively predict its surrounding words. Given a word w and a context w_c, skip-gram tries to maximize the conditional probability $P(w_c|w)$ through a softmax process. Whereas, this approach has a problem. To calculate $P(w_c|w)$, it involves scanning the whole vocabulary, which is usually large. Therefore, the full calculation is computationally expensive. Skip-gram approximates this conditional probability value using negative sampling (NEG) method which is a simplified method from noise contrastive estimation (NCE) [9] method. Our embedding model is an extension based on skip-gram.

Some works have solved EL tasks using neural networks. Huang [2] used a deep neural network (DNN) to train entity embedding and sorted candidate entities using a semi-supervised graph regularization model. Hu [10] improved entity embedding using a structured knowledge which is derived from Wikipedia's catalogue and construct a model to maximize global consistency between predicted entities. Whereas, these approaches learn entities embedding separately and do not interact with words. Yamada [4] proposed a joint learning model that maps word and entity to the same contiguous vector space and then ranked candidate entities using a gradient boosting regression tree (GBRT) [13] model. Chen [5] developed a bilinear joint learning model (BJLM) that mapped word and entity to different distribution spaces, and then used a pairwise boosting regression tree (PBRT) [7] model to evaluate candidate entities. Sun [6] proposed a tensor neural model to imitate the interactions between mentions, contexts, and entities, and then used a local method to sort candidate entities. Francis-Landau

[14] used a convolutional neural network (CNN) to model semantic correspondence between the context of entity mention and candidate entities and then used a logistic regression layer to rank candidate entities. However, neural networks are overly complex and computationally expensive.

Most previous methods have assumed that all the words in a context have equal importance. In contrast, ABJL model considers the diverse contextual impacts of words on the target word or entity so that more fine-grained learning on word and entity embedding can be performed. In addition, textual context features and entity coherence features are also studied via the proposed ABJL model. PBRT [7] is investigated for candidate entities ranking with new features as well as conventional features.

3 Methodology

In this section, our model for joint learning word and entity embedding is proposed. Additionally, the training method of the proposed model is explained in detail.

3.1 Attention-Based Bi-Linear Joint Learning Model

BJLM [5] does not consider the influences of different words on the target word in context that it is a coarse-grained type of learning. To solve the problem, during the BJLM training process, the different effect of each word in the context of target word is considered in our method. Therefore, we propose an attention-based bilinear joint learning model (ABJL) as an extension of BJLM. The so-called attention mechanism addresses different weight for the words in the context. First, word and entity are embedded into different spaces through an initial matrix mapping method. Then the attention mechanism is integrated into the model to calculate the impacts of context words on the target word and entity training. In such as two-stage method, a more elaborate embedding learning on target word or entity is performed. The attention is calculated via Dot-Product method [15] as follows:

$$Attention(C, E) = softmax(\frac{CE^T}{\sqrt{d_k}})E \tag{1}$$

where C is a matrix of all words vector in context. In addition, E is the vector of entity mention or word. In Eq. (1), $C \in R^{n \times d_k}$ and $E \in R^{1 \times d_k}$. In entity linking, C is the context words vector sequence where entity mention located and E is the vector representation of entity mention.

When training the embedding for a word or entity mention, we consider the values of different influences for each word in its context. Formally, given a sequence of N word and entity string s_1, s_2, ..., s_N. ABJL's goal is to maximize the following function:

$$\mathcal{L}_A = \sum_{t=1}^{N} \sum_{s_c \in context(s_i)} \log P_A(s_c|s_i) \tag{2}$$

where S_i represents target string and *context*(s_i) is the context string for s_i. The conditional probability is calculated as follows:

$$P_A(s_c|s_i) = \frac{1}{2}((P_B(s_c|s_i) + Attention(\mathrm{C}_{s_i}, s_c)) \tag{3}$$

where C_{S_i} represents a matrix constructed from vectors of context words of s_i.

The training objective of ABJL is to learn word and entity representations that do best in predicting the nearby words and entities. For example, Fig. 2(a) uses the target word w_t to predict context strings which contain two words (w_{t-1} and w_{t+2}) and an entity e_{t+1}. The attention scores of w_{t-1}, w_{t+2} and e_{t+1} are considered to create a more fine-grained representation of w_t. Figure 2(b) uses the target entity e_t to predict context strings which contain two words (w_{t-1} and w_{t+2}) and an entity e_{t+1}. Again, the attention scores of w_{t-1}, w_{t+2} and e_{t+1} are considered to create a more fine-grained representation of e_t. The projection matrix M is used to bridge the space gap when the target string and the context string are in different embedding types; in contrary, when the target and the context exist in the same embedding types, projection matrix M becomes an identity matrix and does nothing.

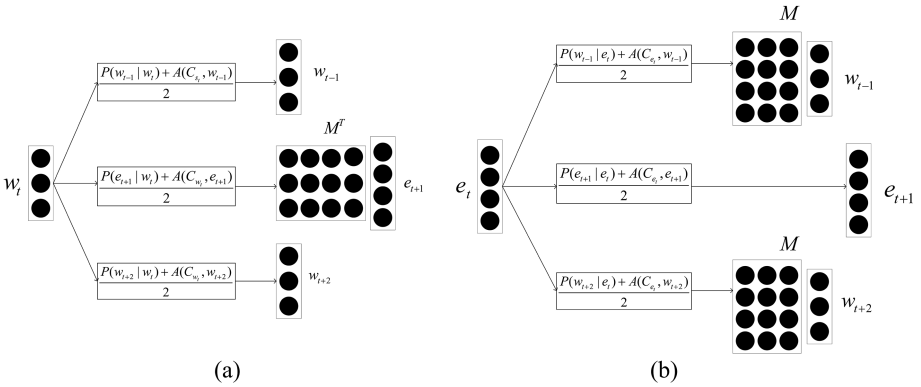

Fig. 2. An example of an ABJL.

3.2 Training

Maximize the function Eq. (2) is the training objective of the proposed model and the result matrix V is used to embed word and entity. One problem is that the computational cost of normalizers contained in $P_A(S_C|S_i)$ is hugely expensive that occurs when training the model because they involve calculating all the words and entities. To solve this problem, negative sampling (NEG) [8] is used to transform the original objective function into a computationally flexible objective function. NEG is defined as follows:

$$\log \sigma(V_{w_t}^T U_{w_{t+j}}) + \sum_{i=1}^{g} E_{w_i} \sim P_{neg(w)}[\log \sigma(-V_{w_t}^T U_{w_i})] \tag{4}$$

where $\sigma(x) = 1/(1 + \exp(-x))$, and g represents the number of negative samples. Equation (4) is used instead of $log(P_A(S_C|S_i))$ in Eq. (2). Therefore, the objective function Eq. (2) is transformed into a simple binary classification objective function, which distinguishes observed word w_t from the word extracted from the noise $P_{neg(w)}$. Wikipedia is used to train the proposed model and stochastic gradient descent (SGD) [16] is applied for optimization. Maximize the transformed objective function by iterating over Wikipedia page multiple times.

4 Entity Linking Using Embedding

In this section, how to apply the proposed embedding model to EL task is explained in detail. First, a formal definition of EL is given: Given a knowledge base, the goal of EL is to match each entity mention to its corresponding entity in the knowledge base. Note that a named entity mention is a token sequence in a text. The mention may refer to an entity and is pre-identified. EL task usually consists of two subtasks: generation of candidate entities and ranking of candidate entities. Thus, we also discuss the candidate entity ranking.

The key to improving ranking performance is effectively modeling context by entity mention. In Sects. 4.1 and 4.2, two new methods for modeling contexts are modified by using the proposed embedding method. Furthermore, these two models are combined with the traditional EL features [1] as input features for sorting model in following Sect. 4.3.

4.1 Modeling Textual Context Information

The design of textual context feature is based on the assumption that if a given mention and an entity have a similar context, then that the mention and entity are more likely refer the same thing. Yamada [4] proposed an approach that used embedding to measure the similarity between textual context and entity. First, it derives the vector representation of the textual context. Second, it uses cosine similarity to calculate the similarity between text context and entity.

When calculating the context vector, Yamada simply uses an averaging method to sum the word vectors in context. This ignores the greater impact of the more important words in the context and elevates the importance of unimportant words to the average, which is obviously unreasonable. To avoid the problem, the cosine similarity in this paper is calculated between the word in the context and target entity instead of in this stage, where the importance of the entity is determined by the cosine similarity. The textual context vector is derived by weighted summation of the context word vector:

$$\overrightarrow{v_{c_m}} = \sum_{w \in W_{cm}} \frac{a_{c_{m_i}}}{\sum_1^M a_{c_{m_j}}} \overrightarrow{v_{m_i}} \tag{5}$$

where W_{c_m} is a set of entity mention's context words. M denotes the size of the set, $\overrightarrow{v_{m_i}} \in V$ is the vector representation of word w, and $a_{c_{m_i}}$ represents the similarity between the i_{th} words in context and target entity mention which is calculated by cosine similarity.

Next, similarity between each candidate and the obtained textual context is calculated, which is obtained by calculating the cosine similarity of textual context vector $\overrightarrow{v_{c_m}}$ and the entity vector $\overrightarrow{v_e}$.

4.2 Modeling Entity Coherence

Milne [17] found that effectively modeling coherence of entities is certainly important for assigning entities to mentions in EL. Since most texts deal with one or several semantically related topics, entity consistency becomes a key metric, such as rock music, Internet technology, or global warming. However, not all content is together.

We use a simple two-step approach [18] to solve this problem: First, a coherence score is used to train machine learning models that are among unambiguous mentions. Then, in the second step, the model is retrained using the coherence score between predicted entities. To calculate entity coherence value, our method computes context entities vector, and then measures the similarity between context entities vector and target entity vector. It should be noted that context entities in the first step are unambiguous entities, while the second step uses the predicted entities. Based on this idea, the context entity vector is derived via a weighted summation using cosine similarity:

$$\overrightarrow{v_{c_e}} = \sum_{e \in E_{c_m}} \frac{a_{c_{e_i}}}{\sum_1^N a_{c_{e_j}}} \overrightarrow{v_{e_i}} \tag{6}$$

where E_{c_m} represents the set of entities in the context of m, N is the size of the set, and $a_{c_{e_i}}$ represents the similarity between the i_{th} entity in context and target entity mention.

4.3 Pairwise Ranking Model

Given an entity mention, a ranking score is assigned to each candidate entity by the ranking model. EL system selects the candidate entity with the highest score as the referential entity. Shen [19] regarded EL as a pairwise ranking problem and presented a method based on SVM ranking [16]. Yamada [4] ranked candidate entities using a GBRT with a pointwise loss function. Yet, the pointwise sorting method may cause label deviation problems because there are many candidate entities for a given mention but only one is correct. However, this paper does not intend to study various ranking methods for EL. In our work, a supervised PBRT [7] model is adapted to rank candidate entities.

5 Experiments

In this section, the experimental settings and result are discussed. First, the training method and training tools are explained. Then, the experimental details on two standard EL data sets are introduced. At last, the experimental result is comprehensively analyzed. To demonstrate the effectiveness of the proposed model, we compared the accuracy with those of other state-of-the-art methods on the CoNLL and TAC 2010 datasets.

5.1 Prerequisites

To train our proposed model, Wikipedia dump of September 2018 version is used. The dump file is parsed with JWPL [20], where navigation, maintenance, discussion, redirected and disambiguated pages are initially removed. All page titles and anchors from each page are extracted as reference entities. Through Wikipedia links, the anchor on the page is replaced with the title of the page that it points to.

In the experiments, the dimension size of embedding is 300, the size of the context window c is 10 and the number of negative samples g is 20. Learning rate α is 0.025 and is linearly decreased during the iteration. The model iterates all pages in Wikipedia dump 10 times online.

For the CoNLL dataset, mentions only with legal corresponding entities in the knowledge base are selected. Standard micro-accuracy which aggregates over all mentions and macro-accuracy which aggregates over all documents are used for the measurements of the algorithm. For TAC2010 dataset, the preprocess is the same as CoNLL dataset but only micro-accuracy is used.

Evaluation Metrics. *Precision, recall, F1-measure*, and *accuracy* are usually used as the evaluation metrics to perform the assessments of EL systems. The *precision* of the entity linking system is calculated as the percentage of correctly linked mentions that are generated by the system [1]:

$$precision = \frac{|\{correctly\ linked\ entity\ mentions\}|}{|\{linked\ mentions\ generated\ by\ system\}|} \tag{7}$$

Precision considers all entity mentions linked by the system and determines the percentage of correct entity mentions linked by the EL system. *Precision* is usually used in conjunction with the *recall* metric, which is the fraction of correctly linked entity mentions that should be linked [1]:

$$recall = \frac{|\{correctly\ linked\ entity\ mentions\}|}{|\{entity\ mentions\ that\ should\ be\ linked\}|} \tag{8}$$

Recall considers all the entity mentions that should be linked and determines how correctly linked entity mentions are with regard to total entity mentions that should be linked. These two measures are sometimes used together in F_1*-measure* to provide a single measurement for a system. F_1*-measure* is defined as the harmonic mean of *precision* and *recall* [1]:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{9}$$

For our experiment, entity mentions that should be linked are provided as the input of EL system; consequently, the number of linked mentions generated by the experiment always equals the number of entity mentions that should be linked. In this situation, researchers usually use *accuracy* to assess the system's performance. *Accuracy* is calculated as the number of correctly linked entity mentions divided by the total number of all entity mentions. Therefore, here *precision* = *recall* = F_1 = *accuracy*. Moreover, accuracy is also regarded as the official evaluation measure in the TAC-KBP track.

5.2 Entity Link

Set Up. We test our proposed model's performance on two standard EL datasets: The CoNLL dataset and the TAC 2010 dataset. The details of the two data sets are described below.

CoNLL. The CoNLL dataset is constructed by Hoffart [3] which is a popular EL dataset. The CoNLL dataset includes three parts: training, development, and test sets. The training set is used to train our learning model and the performance of our approach is measured using the test set. In the CoNLL dataset, each mention is annotated with an entity.

TAC 2010. The TAC 2010 dataset is another popular EL dataset. The dataset was constructed by Ji [21] for the Text Analysis Conference (TAC). This data set was constructed based on news articles from various proxy and weblog data, and it contains two collections: the training set and test set. We only use entity mentions where a matching valid entity exists in the knowledge base. The training set is adopted to train our model. And the test set is used to evaluate its performance. In most included documents, a query mention has been annotated with an entity.

Baseline Methods. We compare our method with the following recently proposed state-of-the-art methods:

- Globerson [12] proposed a coherence model with a multi-focal attention mechanism.
- PPRsim [11] is a graph-based EL approach based on Personalized PageRank.
- Yamada [4] presented a joint embedding model and utilized a GBRT model to rank candidate entities.
- Chen [5] developed a bilinear joint learning model and utilized a PBRT model to rank candidate entities.

Knowledge Base and Candidate Entity Generation. We used the Wikipedia of September 2018 version as our reference database. Wikipedia is a free, online, decentralized, multi-language encyclopedia that was created by thousands of volunteers from all over the world. In Wikipedia, each basic entry is an article. The article defines and describes an entity or a topic. And each article is uniquely referenced by an identifier. Besides, Wikipedia has high coverage of named entities and contains a wealth of knowledge for well-known entities. In addition, a rich set of features is provided by the structure of Wikipedia for entity linking. The features include article directories, entity pages, disambiguation pages, redirect pages, and hyperlinks in Wikipedia articles. These features are highly beneficial for EL tasks.

The way we construct a set of candidate entities for mentions that appear in the TAC 2010 dataset is to construct a candidate entity dictionary. We use the title of Wikipedia entity page and the text of bold font in the first paragraph to construct the dictionary. Then we use all the anchors as keys and the corresponding Wikipedia titles as values to construct a key-value dictionary. Finally, we use this dictionary to generate candidate entities. To perform candidate generation for the CONLL data sets, we use a publicly available third-party dictionary [11].

5.3 Experimental Result

In this section, we analyze our experimental results, the prediction errors, and features. A detailed analysis of the above aspects for CoNLL data set is conducted.

$PBRT_A$ refers to the proposed model are all the models are trained with the features derived from ABJL. Table 1 shows the experimental results of our model and the compared baseline model in two data sets. $PBRT_A$ achieves a micro-precision of 0.947 and a macro-precision of 0.943 on CoNLL dataset and achieves a micro-precision of 0.893 on TAC-KBP 2010 dataset. $PBRT_A$ performs better than the baselines on both datasets.

Table 1. Accuracy scores on CoNLL and TAC-KBP 2010 datasets

	CoNLL (micro)	CoNLL (macro)	TAC2010 (micro)
$PBRT_A$	**0.947**	**0.943**	**0.893**
Chen	0.938	0.935	0.881
Yamada	0.931	0.926	0.855
PPRSim	0.918	0.899	-
Globerson	0.927	-	0.872

Our model not only achieves good experimental results but results that are statistically significant. On the CoNLL and TAC datasets, our model achieves advanced EL results and improves the accuracy of EL. This result occurs because ABJL constructs more accurate representations of words and entities. When ABJL trains word and entity embeddings, it considers the different impacts of the words and entities in the context which causes the trained embedding to more accurately represent the semantics of

words and entities. More accurate embeddings yield more realistic results when calculating the similarity between words and between entities.

In the experiments, we primarily encountered the following two typical errors. The first type error is one of common sense. For example, when the context is limited, a country name that appears after another person's name usually refers to a country. One sentence "*Warcraft-17-Jack Stimulus (America) played very well*". The mention *America* has two candidate entities, the *National Football League* and the *United States*. We can infer that the intent of the sentence is nationality when *America* comes after a name. The second type error is also a common sense error but is more difficult to correct. For example, in the sentence "*Santa Fe has mining and mining operations in Nevada, California, Montana, Canada, Brazil, Australia, Chile, Kazakhstan, Mexico and Ghana.*" [5], when no additional information is available, it is also difficult for people to understand whether *Mexico* means the country of *Mexico* or the *city of Mexico*. Solving the preceding types of errors is a difficult challenge. Understanding sentences by applying real-world common sense is a more appropriate approach.

6 Conclusion

In this paper, we proposed a new bilinear joint learning model with an attention mechanism (ABJL). When ABJL trains target word or entity embeddings, it expresses the target word and entity more finely and accurately by capturing the various influences and contributions of the contextual words around the target. Embedding trained by the proposed model is used to construct two types of features that are inputted into the PBRT along with traditional EL features. An excellent result is achieved on two standard EL databases showing that ABJL produces efficient embedding that improve the performance of our EL method.

In future work, we plan to further study the application of our model on large-scale datasets and to evaluate the use of distributed clustering methods to address the challenges of large-scale datasets. In addition, because it is still a challenge for computers to acquire real-world common sense knowledge, we want to apply real-world common sense and its relations in a knowledge graph to improve our model.

Acknowledgements. This work is supported by the National Key Research and Development Plan of China under Grant No. 2017YFD0400101, the Natural Science Foundation of Shanghai under Grant No. 16ZR1411200, and the CERNET Innovation Project under Grant No. NGII20170513.

References

1. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2015)
2. Huang, H., Heck, L., Ji, H.: Leveraging deep neural networks and knowledge graphs for entity disambiguation. arXiv preprint arXiv:1504.07678 (2015)

3. Hoffart, J., Yosef, M.A., Bordino, I., et al.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 782–792. Association for Computational Linguistics (2011)
4. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. arXiv preprint arXiv:1601.01343 (2016)
5. Chen, H., Wei, B., Liu, Y., Li, Y., Yu, J., Zhu, W.: Bilinear joint learning of word and entity embeddings for entity linking. Neurocomputing **294**, 12–18 (2018)
6. Sun, Y., Lin, L., Tang, D., et al.: Modeling mention, context and entity with neural networks for entity disambiguation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 632–639 (2015)
7. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
8. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
9. Gutmann, M.U., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. J. Mach. Learn. Res. **13**, 307–361 (2012)
10. Hu, Z., Huang, P., Deng, Y., et al.: Entity hierarchy embedding. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1292–1300 (2015)
11. Pershina, M., He, Y., Grishman, R.: Personalized page rank for named entity disambiguation. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 238–243 (2015)
12. Globerson, A., Lazic, N., Chakrabarti, S., et al.: Collective entity resolution with multi-focal attention. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 621–631 (2016)
13. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
14. Francis-Landau, M., Durrett, G., Klein, D.: Capturing semantic similarity for entity linking with convolutional neural networks. arXiv preprint arXiv:1604.00734 (2016)
15. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Laporte, L., Flamary, R., Canu, S., et al.: Nonconvex regularizations for feature selection in ranking with sparse SVM. IEEE Trans. Neural Netw. Learn. Syst. **25**(6), 1118–1130 (2013)
17. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518. ACM (2008)
18. Ratinov, L., Roth, D., Downey, D., et al.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 1375–1384. Association for Computational Linguistics (2011)
19. Shen, W., Wang, J., Luo, P., et al.: Linden: linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st International Conference on World Wide Web, pp. 449–458. ACM (2012)

20. Ferschke, O., Zesch, T., Gurevych, I.: Wikipedia revision toolkit: efficiently accessing wikipedia's edit history. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations, pp. 97–102. Association for Computational Linguistics (2011)
21. Ji, H., Grishman, R., Dang, H.T., et al.: Overview of the TAC 2010 knowledge base population track. In: Third Text Analysis Conference (TAC 2010), vol. 3, no. 2, pp. 3 (2010)